

**ANALOG/INTERFACE
H A N D B O O K
2000**




MICROCHIP
The Embedded Control Solutions Company®

For:
Abby Kagan
@ Weidmuller



Analog/Interface Handbook 2000

Microchip Technology Inc.
1150 Corporate Park Drive
Chandler, AZ 85248 USA
Telephone: (480) 792-7200
Fax: (480) 792-7277
E-mail: info@microchip.com
http://www.microchip.com
Copyright © 2000 Microchip Technology Inc. All rights reserved.
Microchip is a registered trademark of Microchip Technology Inc.
All other trademarks mentioned herein are property of their respective
owners.

Microchip Technology Inc.
1150 Corporate Park Drive
Chandler, AZ 85248 USA
Telephone: (480) 792-7200
Fax: (480) 792-7277
E-mail: info@microchip.com
http://www.microchip.com
Copyright © 2000 Microchip Technology Inc. All rights reserved.
Microchip is a registered trademark of Microchip Technology Inc.
All other trademarks mentioned herein are property of their respective
owners.



Serial
PIC
Microchip

Endurance
In-Circuit Serial Programming (ICSP)
microID
FilterLab
Serialized Quick Term Programming (SQTP)

"All rights reserved. Copyright © 2000, Microchip Technology Incorporated, USA. Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights."

Trademarks

The Microchip name, logo, KEELOQ, PIC, PICMASTER, PICmicro, PRO MATE, PICSTART, MPLAB, and SEEVAL are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Total Endurance, In-Circuit Serial Programming (ICSP), microID, FilterLab are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2000, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

TO OUR VALUED CUSTOMERS:

Welcome to the *Analog Interface Handbook 2000*. This new handbook includes all application notes, datasheets and product briefs which have been written and published over the past year.

Microchip will continue publishing application notes, technical briefs and reference designs in a series of supplemental handbooks called 'Updates'. These Updates, with revised and new documents, will be incorporated into future volumes as appropriate. As individual application notes become available, they will be posted to our web site for download at: www.microchip.com.

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced. We welcome your feedback.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via email at docerrors@mail.microchip.com or fax at (480) 786-7277.

New Customer Notification System

Register on our web site at www.microchip.com/cn to receive the most current information on our products.



OT RUO VAI NED CUSTOMEER

Microchip Technology Inc. (Microchip) is a leading provider of integrated circuits (ICs) for the microcontroller, analog, memory, and discrete markets. Microchip's products are used in a wide range of applications, including automotive, industrial, medical, and consumer products. Microchip's commitment to quality and reliability has earned it a reputation as a leader in the industry. Microchip's products are manufactured using state-of-the-art manufacturing processes and are backed by a comprehensive support network. Microchip's products are available through a variety of distribution channels, including independent distributors, authorized representatives, and direct sales. Microchip's products are also available through its website at www.microchip.com.

If you have any questions or comments about Microchip products, please contact the Microchip Technical Support Department at (482) 789-5555. Microchip's Technical Support Department is available 24 hours a day, 7 days a week, to answer your questions and provide you with the information you need.

Table of Contents

	PAGE
SECTION 1 MICROCHIP TECHNOLOGY INC. COMPANY PROFILE	SECTION 2 APPLICATION NOTES
1-B-1 Company Profile.....	1-1
1-B-2 CAN Development Kit.....	1-2
1-B-3 CAN Filter Design Tool.....	1-3
SECTION 2 APPLICATION NOTES	SECTION 3 DATASHEETS
2-1 Smart Sensor CAN Node using the MCP2510 and PIC16F876 - AN212	2-1
2-83 Temperature Sensing Technologies - AN679	2-83
2-93 Using Single Supply Operational Amplifiers in Embedded Systems - AN682	2-93
2-101 Single Supply Temperature Sensing with Thermocouples - AN684	2-101
2-117 Thermistors in Single Supply Temperature Sensing Circuits - AN685	2-117
2-127 Understanding and Using Supervisory Circuits - AN686	2-127
2-131 Precision Temperature Sensing with RTD Circuits - AN687	2-131
2-135 Layout Tips for 12-Bit A/D Converter Application - AN688	2-135
2-141 Understanding A/D Converter Performance Specifications - AN693	2-141
2-151 Interfacing Pressure Sensors to Microchip's Analog Peripherals - AN695	2-151
2-161 Anti-Aliasing, Analog Filters for Data Acquisition Systems - AN699	2-161
2-171 Interfacing Microchip MCP3201 A/D Converter to 8051-Based Microcontroller - AN702	2-171
2-183 Using the MCP320X 12-Bit Serial A/D Converter with Microchip PICmicro® Devices - AN703	2-183
2-205 Interfacing Microchip's MCP3201 Analog/Digital (A/D) Converter to MC68HC11E9-Based Microcontroller - AN704	2-205
2-215 Controller Area Network (CAN) Basics - AN713	2-215
2-223 Building a 10-bit Bridge Sensing Circuit using the PIC16C6XX and MCP601 Operational Amplifier - AN717	2-223
2-231 Interfacing Microchip's MCP3201 Analog-to-Digital Converter to the PICmicro® Microcontroller - AN719	2-231
2-251 Operational Amplifier Topologies and DC Specifications - AN722	2-251
2-261 Operational Amplifier AC Specifications and Applications - AN723	2-261
2-271 Using the MCP2510 CAN Developer's Kit - AN733	2-271
SECTION 3 DATASHEETS	
3-1 Microcontroller Supervisory Circuit with Push-Pull Output - MCP100/101	3-1
3-9 Microcontroller Supervisory Circuit with Open Drain Output - MCP120/130	3-9
3-15 Microcontroller Supervisory Circuit with Push-Pull Output - MCP809/810	3-15
3-23 Stand-Alone CAN Controller with SPI® Interface - MCP2510	3-23
3-99 2.7V to 5.5V Single Supply CMOS Op Amps - MCP601/602/603/604	3-99
3-119 2.5V to 5.5V Micropower CMOS Op Amps - MCP606/607/608/609	3-119
3-149 2.7V 10-Bit A/D Converter with SPI™ Serial Interface - MCP3001	3-149
3-167 2.7V Dual Channel 10-Bit A/D Converter with SPI™ Serial Interface - MCP3002	3-167
3-185 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI™ Serial Interface - MCP3004/3008	3-185
3-205 2.7V 12-Bit A/D Converter with SPI® Serial Interface - MCP3201	3-205
3-223 2.7V Dual Channel 12-Bit A/D Converter with SPI® Serial Interface - MCP3202	3-223
3-241 2.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI® Serial Interface - MCP3204/3208	3-241
SECTION 4 PICMICRO® MICROCONTROLLER PRODUCT BRIEFS	
4-1 High-Performance Microcontrollers with 10-Bit A/D - PIC18Cxx2	4-1
4-3 28/40-Pin, 8-Bit CMOS Microcontrollers w/ 12-Bit A/D - PIC16C77X	4-3
4-7 28/40-pin 8-Bit CMOS FLASH Microcontrollers - PIC16F870/871	4-7
4-11 28-Pin, 8-Bit CMOS FLASH Microcontroller - PIC16F872	4-11
4-13 28/40-pin 8-Bit CMOS FLASH Microcontrollers - PIC16F87X	4-13
4-17 18/20-Pin, 8-Bit CMOS Microcontrollers with 10/12-Bit A/D - PIC16C717/770/771	4-17

Table of Contents - Continued

PAGE	PAGE
SECTION 5 DEVELOPMENT SYSTEMS	
1-1 On-Line Support MCP2510 FilterLab™ MXDEV™ 1 1-5 1-6 1-7 1-8 1-9 1-10 1-11 1-12 1-13 1-14 1-15 1-16 1-17 1-18 1-19 1-20 1-21 1-22 1-23 1-24 1-25 1-26 1-27 1-28 1-29 1-30 1-31 1-32 1-33 1-34 1-35 1-36 1-37 1-38 1-39 1-40 1-41 1-42 1-43 1-44 1-45 1-46 1-47 1-48 1-49 1-50 1-51 1-52 1-53 1-54 1-55 1-56 1-57 1-58 1-59 1-60 1-61 1-62 1-63 1-64 1-65 1-66 1-67 1-68 1-69 1-70 1-71 1-72 1-73 1-74 1-75 1-76 1-77 1-78 1-79 1-80 1-81 1-82 1-83 1-84 1-85 1-86 1-87 1-88 1-89 1-90 1-91 1-92 1-93 1-94 1-95 1-96 1-97 1-98 1-99 1-100 1-101 1-102 1-103 1-104 1-105 1-106 1-107 1-108 1-109 1-110 1-111 1-112 1-113 1-114 1-115 1-116 1-117 1-118 1-119 1-120 1-121 1-122 1-123 1-124 1-125 1-126 1-127 1-128 1-129 1-130 1-131 1-132 1-133 1-134 1-135 1-136 1-137 1-138 1-139 1-140 1-141 1-142 1-143 1-144 1-145 1-146 1-147 1-148 1-149 1-150 1-151 1-152 1-153 1-154 1-155 1-156 1-157 1-158 1-159 1-160 1-161 1-162 1-163 1-164 1-165 1-166 1-167 1-168 1-169 1-170 1-171 1-172 1-173 1-174 1-175 1-176 1-177 1-178 1-179 1-180 1-181 1-182 1-183 1-184 1-185 1-186 1-187 1-188 1-189 1-190 1-191 1-192 1-193 1-194 1-195 1-196 1-197 1-198 1-199 1-200 1-201 1-202 1-203 1-204 1-205 1-206 1-207 1-208 1-209 1-210 1-211 1-212 1-213 1-214 1-215 1-216 1-217 1-218 1-219 1-220 1-221 1-222 1-223 1-224 1-225 1-226 1-227 1-228 1-229 1-230 1-231 1-232 1-233 1-234 1-235 1-236 1-237 1-238 1-239 1-240 1-241 1-242 1-243 1-244 1-245 1-246 1-247 1-248 1-249 1-250 1-251 1-252 1-253 1-254 1-255 1-256 1-257 1-258 1-259 1-260 1-261 1-262 1-263 1-264 1-265 1-266 1-267 1-268 1-269 1-270 1-271 1-272 1-273 1-274 1-275 1-276 1-277 1-278 1-279 1-280 1-281 1-282 1-283 1-284 1-285 1-286 1-287 1-288 1-289 1-290 1-291 1-292 1-293 1-294 1-295 1-296 1-297 1-298 1-299 1-300 1-301 1-302 1-303 1-304 1-305 1-306 1-307 1-308 1-309 1-310 1-311 1-312 1-313 1-314 1-315 1-316 1-317 1-318 1-319 1-320 1-321 1-322 1-323 1-324 1-325 1-326 1-327 1-328 1-329 1-330 1-331 1-332 1-333 1-334 1-335 1-336 1-337 1-338 1-339 1-340 1-341 1-342 1-343 1-344 1-345 1-346 1-347 1-348 1-349 1-350 1-351 1-352 1-353 1-354 1-355 1-356 1-357 1-358 1-359 1-360 1-361 1-362 1-363 1-364 1-365 1-366 1-367 1-368 1-369 1-370 1-371 1-372 1-373 1-374 1-375 1-376 1-377 1-378 1-379 1-380 1-381 1-382 1-383 1-384 1-385 1-386 1-387 1-388 1-389 1-390 1-391 1-392 1-393 1-394 1-395 1-396 1-397 1-398 1-399 1-400 1-401 1-402 1-403 1-404 1-405 1-406 1-407 1-408 1-409 1-410 1-411 1-412 1-413 1-414 1-415 1-416 1-417 1-418 1-419 1-420 1-421 1-422 1-423 1-424 1-425 1-426 1-427 1-428 1-429 1-430 1-431 1-432 1-433 1-434 1-435 1-436 1-437 1-438 1-439 1-440 1-441 1-442 1-443 1-444 1-445 1-446 1-447 1-448 1-449 1-450 1-451 1-452 1-453 1-454 1-455 1-456 1-457 1-458 1-459 1-460 1-461 1-462 1-463 1-464 1-465 1-466 1-467 1-468 1-469 1-470 1-471 1-472 1-473 1-474 1-475 1-476 1-477 1-478 1-479 1-480 1-481 1-482 1-483 1-484 1-485 1-486 1-487 1-488 1-489 1-490 1-491 1-492 1-493 1-494 1-495 1-496 1-497 1-498 1-499 1-500 1-501 1-502 1-503 1-504 1-505 1-506 1-507 1-508 1-509 1-510 1-511 1-512 1-513 1-514 1-515 1-516 1-517 1-518 1-519 1-520 1-521 1-522 1-523 1-524 1-525 1-526 1-527 1-528 1-529 1-530 1-531 1-532 1-533 1-534 1-535 1-536 1-537 1-538 1-539 1-540 1-541 1-542 1-543 1-544 1-545 1-546 1-547 1-548 1-549 1-550 1-551 1-552 1-553 1-554 1-555 1-556 1-557 1-558 1-559 1-560 1-561 1-562 1-563 1-564 1-565 1-566 1-567 1-568 1-569 1-570 1-571 1-572 1-573 1-574 1-575 1-576 1-577 1-578 1-579 1-580 1-581 1-582 1-583 1-584 1-585 1-586 1-587 1-588 1-589 1-590 1-591 1-592 1-593 1-594 1-595 1-596 1-597 1-598 1-599 1-600 1-601 1-602 1-603 1-604 1-605 1-606 1-607 1-608 1-609 1-610 1-611 1-612 1-613 1-614 1-615 1-616 1-617 1-618 1-619 1-620 1-621 1-622 1-623 1-624 1-625 1-626 1-627 1-628 1-629 1-630 1-631 1-632 1-633 1-634 1-635 1-636 1-637 1-638 1-639 1-640 1-641 1-642 1-643 1-644 1-645 1-646 1-647 1-648 1-649 1-650 1-651 1-652 1-653 1-654 1-655 1-656 1-657 1-658 1-659 1-660 1-661 1-662 1-663 1-664 1-665 1-666 1-667 1-668 1-669 1-670 1-671 1-672 1-673 1-674 1-675 1-676 1-677 1-678 1-679 1-680 1-681 1-682 1-683 1-684 1-685 1-686 1-687 1-688 1-689 1-690 1-691 1-692 1-693 1-694 1-695 1-696 1-697 1-698 1-699 1-700 1-701 1-702 1-703 1-704 1-705 1-706 1-707 1-708 1-709 1-710 1-711 1-712 1-713 1-714 1-715 1-716 1-717 1-718 1-719 1-720 1-721 1-722 1-723 1-724 1-725 1-726 1-727 1-728 1-729 1-730 1-731 1-732 1-733 1-734 1-735 1-736 1-737 1-738 1-739 1-740 1-741 1-742 1-743 1-744 1-745 1-746 1-747 1-748 1-749 1-750 1-751 1-752 1-753 1-754 1-755 1-756 1-757 1-758 1-759 1-760 1-761 1-762 1-763 1-764 1-765 1-766 1-767 1-768 1-769 1-770 1-771 1-772 1-773 1-774 1-775 1-776 1-777 1-778 1-779 1-780 1-781 1-782 1-783 1-784 1-785 1-786 1-787 1-788 1-789 1-790 1-791 1-792 1-793 1-794 1-795 1-796 1-797 1-798 1-799 1-800 1-801 1-802 1-803 1-804 1-805 1-806 1-807 1-808 1-809 1-810 1-811 1-812 1-813 1-814 1-815 1-816 1-817 1-818 1-819 1-820 1-821 1-822 1-823 1-824 1-825 1-826 1-827 1-828 1-829 1-830 1-831 1-832 1-833 1-834 1-835 1-836 1-837 1-838 1-839 1-840 1-841 1-842 1-843 1-844 1-845 1-846 1-847 1-848 1-849 1-850 1-851 1-852 1-853 1-854 1-855 1-856 1-857 1-858 1-859 1-860 1-861 1-862 1-863 1-864 1-865 1-866 1-867 1-868 1-869 1-870 1-871 1-872 1-873 1-874 1-875 1-876 1-877 1-878 1-879 1-880 1-881 1-882 1-883 1-884 1-885 1-886 1-887 1-888 1-889 1-890 1-891 1-892 1-893 1-894 1-895 1-896 1-897 1-898 1-899 1-900 1-901 1-902 1-903 1-904 1-905 1-906 1-907 1-908 1-909 1-910 1-911 1-912 1-913 1-914 1-915 1-916 1-917 1-918 1-919 1-920 1-921 1-922 1-923 1-924 1-925 1-926 1-927 1-928 1-929 1-930 1-931 1-932 1-933 1-934 1-935 1-936 1-937 1-938 1-939 1-940 1-941 1-942 1-943 1-944 1-945 1-946 1-947 1-948 1-949 1-950 1-951 1-952 1-953 1-954 1-955 1-956 1-957 1-958 1-959 1-960 1-961 1-962 1-963 1-964 1-965 1-966 1-967 1-968 1-969 1-970 1-971 1-972 1-973 1-974 1-975 1-976 1-977 1-978 1-979 1-980 1-981 1-982 1-983 1-984 1-985 1-986 1-987 1-988 1-989 1-990 1-991 1-992 1-993 1-994 1-995 1-996 1-997 1-998 1-999 1-1000 1-1001 1-1002 1-1003 1-1004 1-1005 1-1006 1-1007 1-1008 1-1009 1-1010 1-1011 1-1012 1-1013 1-1014 1-1015 1-1016 1-1017 1-1018 1-1019 1-1020 1-1021 1-1022 1-1023 1-1024 1-1025 1-1026 1-1027 1-1028 1-1029 1-1030 1-1031 1-1032 1-1033 1-1034 1-1035 1-1036 1-1037 1-1038 1-1039 1-1040 1-1041 1-1042 1-1043 1-1044 1-1045 1-1046 1-1047 1-1048 1-1049 1-1050 1-1051 1-1052 1-1053 1-1054 1-1055 1-1056 1-1057 1-1058 1-1059 1-1060 1-1061 1-1062 1-1063 1-1064 1-1065 1-1066 1-1067 1-1068 1-1069 1-1070 1-1071 1-1072 1-1073 1-1074 1-1075 1-1076 1-1077 1-1078 1-1079 1-1080 1-1081 1-1082 1-1083 1-1084 1-1085 1-1086 1-1087 1-1088 1-1089 1-1090 1-1091 1-1092 1-1093 1-1094 1-1095 1-1096 1-1097 1-1098 1-1099 1-1100 1-1101 1-1102 1-1103 1-1104 1-1105 1-1106 1-1107 1-1108 1-1109 1-1110 1-1111 1-1112 1-1113 1-1114 1-1115 1-1116 1-1117 1-1118 1-1119 1-1120 1-1121 1-1122 1-1123 1-1124 1-1125 1-1126 1-1127 1-1128 1-1129 1-1130 1-1131 1-1132 1-1133 1-1134 1-1135 1-1136 1-1137 1-1138 1-1139 1-1140 1-1141 1-1142 1-1143 1-1144 1-1145 1-1146 1-1147 1-1148 1-1149 1-1150 1-1151 1-1152 1-1153 1-1154 1-1155 1-1156 1-1157 1-1158 1-1159 1-1160 1-1161 1-1162 1-1163 1-1164 1-1165 1-1166 1-1167 1-1168 1-1169 1-1170 1-1171 1-1172 1-1173 1-1174 1-1175 1-1176 1-1177 1-1178 1-1179 1-1180 1-1181 1-1182 1-1183 1-1184 1-1185 1-1186 1-1187 1-1188 1-1189 1-1190 1-1191 1-1192 1-1193 1-1194 1-1195 1-1196 1-1197 1-1198 1-1199 1-1200 1-1201 1-1202 1-1203 1-1204 1-1205 1-1206 1-1207 1-1208 1-1209 1-1210 1-1211 1-1212 1-1213 1-1214 1-1215 1-1216 1-1217 1-1218 1-1219 1-1220 1-1221 1-1222 1-1223 1-1224 1-1225 1-1226 1-1227 1-1228 1-1229 1-1230 1-1231 1-1232 1-1233 1-1234 1-1235 1-1236 1-1237 1-1238 1-1239 1-1240 1-1241 1-1242 1-1243 1-1244 1-1245 1-1246 1-1247 1-1248 1-1249 1-1250 1-1251 1-1252 1-1253 1-1254 1-1255 1-1256 1-1257 1-1258 1-1259 1-1260 1-1261 1-1262 1-1263 1-1264 1-1265 1-1266 1-1267 1-1268 1-1269 1-1270 1-1271 1-1272 1-1273 1-1274 1-1275 1-1276 1-1277 1-1278 1-1279 1-1280 1-1281 1-1282 1-1283 1-1284 1-1285 1-1286 1-1287 1-1288 1-1289 1-1290 1-1291 1-1292 1-1293 1-1294 1-1295 1-1296 1-1297 1-1298 1-1299 1-1300 1-1301 1-1302 1-1303 1-1304 	

Subject Index

	<u>PAGE</u>
The following is an alphabetical subject index for the application notes, that are available in the <i>Analog Interface Handbook 2000</i> .	
Miscellaneous	
12-bit ADC:	
AN693	2-141
AN704	2-205
AN719	2-231
8051 Interface: AN702	2-171
A	
Analog-to-Digital (A/D) Converter:	
AN684	2-101
AN685	2-117
AN687	2-131
AN688	2-135
AN693	2-141
AN699	2-161
AN702	2-171
AN703	2-183
AN704	2-205
AN719	2-231
Anti-Aliasing filter: AN699	2-161
Automotive: AN713	2-215
B	
Bridge Sensor: AN717	2-223
Brown-out detect (BOD): AN686	2-127
C	
CAN Controller:	
AN212	2-1
CAN: AN713	2-215
D	
Difference Amplifier: AN682	2-93
F	
Filter:	
Analog:	
AN682	2-93
AN699	2-161
FilterLab™	5-5
FilterLab™: AN699	2-161
Frames: AN713	2-215
I	
Industrial: AN713	2-215
Instrumentation Amplifier: AN682	2-93
L	
Low Pass Filter: AN699	2-161
M	
MC68HC11: AN704	2-205
MCP130:	
AN704	2-205
AN719	2-231
MCP2510 - CAN Development Kit	5-3
MCP2510:	
AN212	2-1
MCP320x: AN703	2-183
MXDEV™ Analog Evaluation System	5-7
N	
Noise:	
AN688	2-135
AN717	2-223
O	
On-Line Support	5-1
Operational Amplifier:	
AN682	2-93
AN684	2-101
AN685	2-117
AN687	2-131
AN699	2-161
AN717	2-223
AN722	2-251
P	
PCB Layout: AN688	2-135
PIC16C62A: AN703	2-183
PIC16C67: AN719	2-231
PIC16F876:	
AN212	2-1
PICDEM-2: AN719	2-231
Photo Detector Pre-Amp: AN682	2-93
Power on reset (POR): AN686	2-127
Protocol: AN713	2-215
R	
RTD:	
AN679	2-83
AN687	2-131



MICROCHIP

Subject Index - Continued

SECTION 1

MICROCHIP TECHNOLOGY INC.

COMPANY PROFILE

Company Profile 1-1



COMPANY PROFILE

MICROCHIP TECHNOLOGY INC.

SECTION 1



MICROCHIP

Microchip Technology Inc.

Company Profile

The Embedded Control Solutions Company®

Since its inception, Microchip Technology has focused its resources on delivering innovative semiconductor products to the global embedded control marketplace. To do this, we have focused our technology, engineering, manufacturing and marketing resources on synergistic product lines: PICmicro® microcontrollers (MCUs), high-endurance Serial EEPROMs, an expanding product portfolio of analog/interface products, RFID tags and KEELOQ® security devices – all aimed at delivering comprehensive, high-value embedded control solutions to a growing base of customers.

Inside Microchip Technology you will find:

- An experienced executive team focused on innovation and committed to listening to our customers
- A focus on providing high-performance, cost-effective embedded control solutions
- Fully integrated manufacturing capabilities
- A global network of manufacturing and customer support facilities
- A unique corporate culture dedicated to continuous improvement
- Distributor network support worldwide including certified distribution FAEs



Chandler, Arizona: Company headquarters near Phoenix, Arizona; executive offices, R&D and wafer fabrication occupy this 242,000 square-foot multi-building campus.

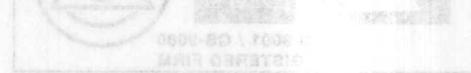
- A Complete Product Solution including:

- RISC OTP, FLASH, EEPROM and ROM MCUs
- A full family of advanced analog MCUs
- KEELOQ security devices featuring patented code hopping technology
- Stand-alone analog and interface products plus microID™ RFID tagging devices
- A complete line of high-endurance Serial EEPROMs
- World-class, easy-to-use development tools
- An Automotive Products Group to engage with key automotive accounts and provide necessary application expertise and customer service

Business Scope

Microchip Technology Inc. designs, manufactures, and markets a variety of CMOS semiconductor components to support the market for cost-effective embedded control solutions.

Microchip's products feature compact size, integrated functionality, ease of development and technical support so essential to timely and cost-effective product development by our customers.



Microchip Technology Inc. is a registered trademark of Microchip Technology Inc. All other trademarks mentioned are the property of their respective owners.



Tempe, Arizona: Microchip's 200,000 square-foot wafer fabrication facility provides increased manufacturing capacity today and for the future.

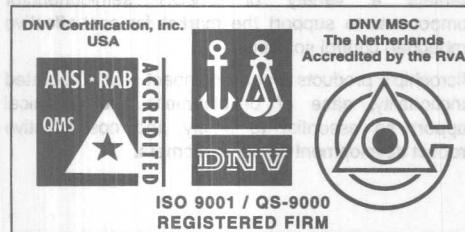
Microchip Technology Inc.

Market Focus

Microchip targets select markets where our advanced designs, progressive process technology and industry-leading product performance enables us to deliver decidedly superior performance. Our Company is positioned to provide a complete product solution for embedded control applications found throughout the consumer, automotive, telecommunication, office automation and industrial control markets. Microchip products are also meeting the unique design requirements of targeted embedded applications including internet, safety and security.

Certified Quality Systems

Microchip's quality systems have been certified to QS-9000 requirements. Its worldwide headquarters and wafer fabrication facilities in Chandler and Tempe, Arizona, received certification on July 23, 1999. The scope of this certification is the design and manufacture of RISC-based MCUs, related non-volatile memory products and microperipheral devices. The quality systems for Microchip's product test facility in Bangkok, Thailand, were QS-9000 certified on February 26, 1999. The scope of this certification is the design and testing of integrated circuits. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



QS-9000 was developed by Chrysler, Ford and General Motors to establish fundamental quality systems that provide for continuous improvement, emphasizing defect prevention and the reduction of variation and waste in the supply chain. Microchip was audited by QS-9000 registrar Det Norske Veritas Certification Inc. of Houston, the same firm which granted Microchip its ISO 9001 Quality System certification in 1997. QS-9000 certification recognizes Microchip's quality systems conform to the stringent standards set forth by the automotive industry, benefiting all customers.

Fully Integrated Manufacturing

Microchip delivers fast turnaround and consistent quality through total control over all phases of production. Research and development, design, mask making, wafer fabrication, and the major part of assembly and quality assurance testing are conducted

at facilities wholly-owned and operated by Microchip. Our integrated approach to manufacturing along with rigorous use of advanced Statistical Process Control (SPC) and a continuous improvement culture has resulted in high and consistent yields which have positioned Microchip as a quality leader in its global markets. Microchip's unique approach to SPC provides customers with excellent pricing, quality, reliability and on-time delivery.



Bangkok, Thailand: Microchip's 200,000 square-foot manufacturing facility houses the technology and assembly/test equipment for high speed testing and packaging.

A Global Network of Plants and Facilities

Microchip is a global competitor providing local services to the world's technology centers. The Company's design and technology advancement facilities, and wafer fabrication sites are located in Chandler and Tempe, Arizona.

The Tempe facility provides an additional 200,000 square feet of manufacturing space that meets the increased production requirements of a growing customer base, and provides production capacity which more than doubles that of Chandler.

Microchip facilities in Bangkok, Thailand, and Shanghai, China, serve as the foundation of Microchip's extensive assembly and test capability located throughout Asia. The use of multiple fabrication, assembly and test sites, with more than 640,000-square-feet of facilities worldwide, ensures Microchip's ability to meet the increased production requirements of a fast growing customer base.

Microchip supports its global customer base from direct sales and engineering offices in Asia, North America, Europe and Japan. Offices are staffed to meet the high quality expectations of our customers, and can be accessed for technical and business support. The Company also franchises more than 60 distributors and a network of technical manufacturer's representatives serving 24 countries worldwide.

Embedded Control Overview

Unlike "processor" applications such as personal computers and workstations, the computing or controlling elements of embedded control applications are embedded inside the application. The consumer is only concerned with the very top-level user interface such as keypads, displays and high-level commands. Very rarely does an end-user know (or care to know) the embedded controller inside (unlike the conscientious PC users, who are intimately familiar not only with the processor type, but also its clock speed, DMA capabilities and so on).

It is, however, most vital for designers of embedded control products to select the most suitable controller and companion devices. Embedded control products are found in all market segments: consumer, commercial, PC peripherals, telecommunications, automotive and industrial. Most embedded control products must meet special requirements: cost effectiveness, low-power, small-footprint and a high level of system integration.

Typically, most embedded control systems are designed around an MCU which integrates on-chip program memory, data memory (RAM) and various peripheral functions, such as timers and serial communication. In addition, these systems usually require complementary Serial EEPROM, analog/interface devices, display drivers, keypads or small displays.

Microchip has established itself as a leading supplier of embedded control solutions. The combination of high-performance PIC12CXXX, PIC16C5X, PIC16CXXX, PIC17CXXX and PIC18CXXX MCU families with Migratable Memory™ technology, along with non-volatile memory products, provide the basis for this leadership. By further expanding our product portfolio to provide precision analog and interface products, Microchip is committed to continuous innovation and improvement in design, manufacturing and technical support to provide the best possible embedded control solutions to you.

PICmicro MCU Overview and Roadmap

Microchip PICmicro MCUs combine high-performance, low-cost, and small package size, offering the best price/performance ratio in the industry. More than one billion of these devices have shipped to customers worldwide since 1990. Microchip offers five families of MCUs to best fit your application needs:

- PIC12CXXX 8-pin 12-bit/14-bit program word
- PIC16C5X 12-bit program word
- PIC16CXXX 14-bit program word
- PIC17CXXX 16-bit program word
- PIC18CXXX enhanced 16-bit program word

All families offer OTP, low-voltage and low-power options, with a variety of package options. Selected members are available in ROM, EEPROM or reprogrammable FLASH versions.

PIC12CXXX: 8-Pin Family

The PIC12CXXX family packs Microchip's powerful RISC-based PICmicro architecture into 8-pin DIP and SOIC packages. These PIC12CXXX products are available with either a 12-bit or 14-bit wide instruction set, a low operating voltage of 2.5V, small package footprints, interrupt handling, a deeper hardware stack, multiple channels and EEPROM data memory. All of these features provide an intelligence level not previously available in applications because of cost or size considerations.

PIC16C5X: 12-Bit Architecture Family

The PIC16C5X is the well-established base-line family that offers the most cost-effective solution. These PIC16C5X products have a 12-bit wide instruction set and are currently offered in 14-, 18-, 20- and 28-pin packages. In the SOIC and SSOP packaging options, these devices are among the smallest footprint MCUs in the industry. Low-voltage operation, down to 2.0V for OTP MCUs, makes this family ideal for battery operated applications. Additionally, the PIC16HV5XX can operate up to 15 volts for use directly with a battery.

PIC16CXXX: 14-Bit Architecture Family

With the introduction of new PIC16CXXX family members, Microchip now provides the industry's highest performance Analog-to-Digital Converter capability at 12-bits for an MCU. The PIC16CXXX family offers a wide-range of options, from 18- to 68-pin packages as well as low to high levels of peripheral integration. This family has a 14-bit wide instruction set, interrupt handling capability and a deep, 8-level hardware stack. The PIC16CXXX family provides the performance and versatility to meet the more demanding requirements of today's cost-sensitive marketplace for mid-range applications.

PIC17CXXX: 16-Bit Architecture Family

The PIC17CXXX family offers the world's fastest execution performance of any MCU family in the industry. The PIC17CXXX family extends the PICmicro MCU's high-performance RISC architecture with a 16-bit instruction word, enhanced instruction set and powerful vectored interrupt handling capabilities. A powerful array of precise on-chip peripheral features provides the performance for the most demanding applications.

PIC18CXXX: 16-Bit Enhanced Architecture Family

The PIC18CXXX is a family of high performance, CMOS, fully static, 16-bit MCUs with integrated analog-to-digital (A/D) converter. All PIC18CXXX MCUs incorporate an advanced RISC architecture. The PIC18CXXX has enhanced core features, 32 level-deep stack, and multiple internal and external interrupt sources. The separate instruction and data busses of the Harvard architecture allow a 16-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches, which require two cycles. A total of 77 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural

Microchip Technology Inc.

innovations used to achieve a very high performance of 10 MIPS for an MCU. The PIC18CXXX family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. These include programmable Low Voltage Detect (LVD) and programmable Brown-Out Detect (BOD).

The Mechatronics Revolution

The nature of the revolution is the momentous shift from analog/electro-mechanical timing and control to digital electronics. It is called the Mechatronics Revolution, and it is being staged in companies throughout the world, with design engineers right on

the front lines: make it smarter, make it smaller, make it do more, make it cost less to manufacture – and make it snappy.

To meet the needs of this growing customer base, Microchip is rapidly expanding its already broad line of PICmicro MCUs. The PIC12CXXX family's size opens up new possibilities for product design.

PICmicro MCU Naming Convention

The PICmicro architecture offers users a wider range of cost/performance options than any MCU family. In order to identify the families, the following naming conventions have been applied to the PICmicro MCUs:

TABLE 1: PICmicro MCU NAMING CONVENTION*

Family	Architectural Features	Name	Technology
PIC18CXXX	8-bit High-Performance MCU Family	PIC18CXX2 PIC18FXXX	OTP program memory with higher resolution analog functions FLASH program memory
	8-bit High-Performance MCU Family	PIC17C4X PIC17CR4X PIC17C7XX	OTP program memory, digital only ROM program memory, digital only OTP program memory with mixed-signal functions
PIC16CXXX	8-bit Mid-Range MCU Family	PIC14CXXX PIC16C55X PIC16C6X PIC16CR6X PIC16C62X PIC16CR62X PIC16CE62X PIC16F62X PIC16C64X PIC16C66X PIC16C7X PIC16CR7X PIC16C7XX PIC16F8X PIC16CR8X PIC16F87X PIC16C9XX	OTP program memory with A/D and D/A functions OTP program memory, digital only OTP program memory, digital only ROM program memory, digital only OTP program memory with comparators OTP program memory with comparators OTP program memory with comparators and EEPROM data memory FLASH program memory with comparators and EEPROM data memory OTP program memory with comparators OTP program memory with comparators OTP program memory with analog functions (i.e. A/D) ROM program memory with analog functions OTP program memory with higher resolution analog functions FLASH program memory and EEPROM data memory ROM program memory and EEPROM data memory FLASH program memory with higher resolution analog functions OTP program memory, LCD driver
	8-bit Base-Line MCU Family	PIC16C5X PIC16CR5X PIC16C505 PIC16HV540	OTP program memory, digital only ROM program memory, digital only OTP program memory, digital only, internal 4 MHz oscillator OTP program memory with high voltage operation
PIC12CXXX	8-bit, 8-pin MCU Family	PIC12C5XX PIC12CE5XX PIC12CR5XX PIC12C67X PIC12CE67X	OTP program memory, digital only OTP program memory, digital only with EEPROM data memory ROM program memory, digital only OTP program memory with analog functions OTP program memory with analog functions and EEPROM data memory

Note 1: The maximum clock speed for some devices is less than 20 MHz.

*Please check with your local Microchip distributor, sales representative or sales office for the latest product information.

Microchip is committed to providing useful and innovative solutions to your embedded system designs. Our installed base of application development systems has grown to an impressive 170,000 systems worldwide.

Among support products offered are MPLAB®-ICE 2000 In-Circuit Emulator running under the Windows® environment. This new real-time emulator supports low-voltage emulation, to 2.0 volts, and full-speed emulation. MPLAB, a complete Integrated Development Environment (IDE), is provided with MPLAB-ICE 2000. MPLAB allows the user to edit, compile and emulate from a single user interface, making the developer productive very quickly. MPLAB-ICE 2000 is designed to provide product development engineers with an optimized design tool for developing target applications. This universal in-circuit emulator provides a complete MCU design toolset for PICmicro MCUs in the PIC12CXXX, PIC16C5X, PIC16CXXX, PIC17CXXX and PIC18CXXX families. MPLAB-ICE 2000 is CE compliant.

Microchip's newest development tool, MPLAB In-Circuit Debugger (ICD) Evaluation Kit, uses the in-circuit debugging capabilities of the PIC16FXXX and PIC18FXXX MCU family and Microchip's ICSPTM capability to debug source code in the application, debug hardware in real time and program a target PIC16FXXX and PIC18FXXX device.

PRO MATE® II, the full-featured, modular device programmer, enables you to quickly and easily program user software into PICmicro MCUs, HCS products and Serial EEPROMs. PRO MATE II runs under MPLAB IDE and operates as a stand-alone unit or in conjunction with a PC-compatible host system.

The PICSTART® Plus development kit is a low-cost development system for the PIC12CXXX, PIC16C5X, PIC16CXXX and PIC17CXXX MCUs.

PICDEM low-cost demonstration boards are simple boards which demonstrate the basic capabilities of the full range of Microchip's MCUs. Users can program the sample MCUs provided with PICDEM boards, on a

easily test firmware. KEELOQ Evaluation Tools support Microchip's HCS Secure Data Products.

The Serial EEPROM Designer's Kit includes everything necessary to read, write, erase or program special features of any Microchip Serial EEPROMs. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

The FilterLab™ Active Filter Design Tool simplifies active filter design for embedded systems designers. The unique FilterLab software automates the design of the anti-aliasing filter for an analog-to-digital converter-based data acquisition system. FilterLab also provides full schematic diagrams of the filter circuit with component values, a SPICE model, and displays the frequency and phase response.

In addition to the FilterLab Active Filter Design Tool, Microchip offers a second analog development tool, the MXDEV™1 Analog Evaluation System, making it easier for embedded systems designers to evaluate and develop with Microchip's line of stand-alone analog products. The hardware and software within the MXDEV 1 system is configured device-specific and allows single or continuous conversions of the analog-to-digital converter under evaluation.

The MCP2510 Controller Area Network (CAN) Developer's Kit makes software developing easy by using a variety of features to manipulate the functionality of the MCP2510. The MCP2510 CAN Developer's kit provides the ability to read, display and modify all registers of the MCP2510 on a bit-by-bit or a byte-by-byte basis.

The microID™ Developer's Kit is an easy-to-use tool for design engineers at all skill levels. Available in a variety of configurations, the microID family of RFID tags can be configured to match existing tags and be directly installed - upgrading to contactless programmability at no added cost. This kit includes all the hardware, software, reference designs and samples required to get started in RFID designs.

TABLE 2: PICmicro SYNERGISTIC DEVELOPMENT TOOLS

Development Tool	Name	PIC12CXXX	PIC16C5X	PIC16CXXX	PIC16F87X	PIC17CXXX	PIC18CXXX
Integrated Development Environment (IDE)	MPLAB™	✓	✓	✓	—	✓	✓
C Compiler	MPLAB-C17	—	—	—	—	✓	—
C Compiler	MPLAB-C18	—	—	—	—	—	✓
Full-Featured, Modular In-Circuit Emulator	MPLAB-ICE 2000	✓	✓	✓	—	✓	✓
In-Circuit Debugger Evaluation Kit	MPLAB-ICD	—	—	—	✓	—	—
Full-Featured, Modular Device Programmer	PRO MATE® II	✓	✓	✓	—	✓	✓
Entry-Level Development Kit with Programmer	PICSTART® Plus	✓	✓	✓	—	✓	✓

Microchip Technology Inc.

Software Support

MPLAB Integrated Development Environment (IDE) is a Windows-based development platform for Microchip's PICmicro MCUs. MPLAB IDE offers a project manager and program text editor, a user-configurable toolbar containing four pre-defined sets and a status bar which communicates editing and debugging information.

MPLAB-IDE is the common user interface for Microchip development systems tools including MPLAB Editor, MPASM Assembler, MPLAB-SIM Software Simulator, MPLIB, MPLINK, MPLAB-C17 Compiler, MPLAB-C18 Compiler, MPLAB-ICE 2000, PRO MATE II Programmer and PICSTART Plus Development Programmer.

Microchip endeavors at all times to provide the best service and responsiveness possible to its customers. The Microchip Internet site can provide you with the latest technical information, production released software for development tools, application notes and promotional news on Microchip products and technology. The Microchip World Wide Web address is <http://www.microchip.com>.

Secure Data Products Overview

Microchip's patented KEELOQ® code hopping technology is the perfect solution for remote keyless entry and logical/physical access control systems. The initial device in the family, the HCS300 encoder, replaces current fixed code encoders in transmitter applications providing a low cost, integrated solution. The KEELOQ family is continuing to expand with the HCS301 (high voltage encoder), HCS200 (low-end, low-cost encoder), and high-end encoders (HCS360 and HCS361) that meet OEM specifications and requirements. The HCS410, a self-powered transponder superset of the HCS360, is the initial device in a new and expanding encoder/transponder family.

Microchip provides flexible decoder solutions by providing optimized routines for Microchip's PICmicro MCUs. This allows the designer to combine the decoder and system functionality in a MCU. The decoder routines are available under a license agreement. The HCS500, HCS512 and HCS515 are the first decoder devices in the KEELOQ family. These devices are single chip decoder solutions and simplify designs by handling learning and decoding of transmitters.

The KEELOQ product family is expanding to include enhanced encoders and decoders. Typical applications include automotive RKE, alarm and immobilizer systems, garage door openers and home security systems.



KEELOQ Encoder Devices

Product	Transmission Code Length Bits	Code Hopping Bits	Prog. Encryption Key Bits	Seed Length	Operating Voltage
HCS101*	66	—	—	—	3.5V to 13.0V
HCS200	66	32	64	32	3.5V to 13.0V
HCS201*	66	32	64	32	3.5V to 13.0V
HCS300	66	32	64	32	2.0V to 6.3V
HCS301	66	32	64	32	3.5V to 13.0V
HCS320	66	32	64	32	3.5V to 13.0V
HCS360	67	32	64	48	2.0V to 6.6V
HCS361	67	32	64	48	2.0V to 6.6V
HCS365*	69	32	2 x 64	60	2.0V to 6.6V
HCS370*	69	32	2 x 64	60	2.0V to 6.6V
HCS410	69	32	64	60	2.0V to 6.6V
HCS412*	69	32	64	60	2.0V to 6.6V
HCS470*	69	32	2 x 64	60	2.0V to 6.6V

KEELOQ Decoder Devices

Product	Reception Length Bits	Transmitters Supported	Functions	Operating Voltage
HCS500	67	Up to 7	15 Serial Functions	4.5V to 5.5V
HCS512	67	Up to 4	15 (S0, S1, S2, S3); V _{LOW} , Serial	3.0V to 6.0V
HCS515	67	Up to 7	15 Serial; 3 Parallel	4.5V to 5.5V

*Contact Microchip Technology Inc. for availability.

Analog/Interface Products

Using its technology achievements in developing analog circuitry for its PICmicro MCU family, the Company launched a complementary line of stand-alone analog and interface products. Many of these stand-alone devices support functionality that may not currently available on PICmicro MCUs. Stand-alone analog IC products currently offered include:

- Analog-to-Digital Converters
- Operational Amplifiers
- System Supervisors

Microchip also offers innovative silicon products to support a variety of bus interfaces used to transmit data to and from embedded control systems. The first interface products support Controller Area Network (CAN), a bus protocol highly integrated into a variety of networked applications including automotive.

High-Performance 12-Bit Analog-to-Digital Converters

The MCP320X 12-bit analog-to-digital converter (ADC) family is based on a successive approximation register architecture. The first four members include: MCP3201, MCP3202, MCP3204 and MCP3208. The MCP320X family features 100K samples per second throughput, low power of 400 microamps active and 500 nanoamps standby, wide supply voltage of 2.7-5.5 volts, extended industrial temperature range of -40° to 85°, +/- 1 LSB DNL and +/- 1 LSB INL max. at 100 ksp/s, no missing codes, and a serial output with an industry-standard SPI™ bus interface. The MCP320X is available in 1-, 2-, 4-, and 8-input channel versions (the MCP3201, MPC3202, MCP3204 and MCP3208, respectively). The devices

are offered in PDIP, SOIC and TSSOP packages. Applications include data acquisition, instrumentation and measurement, multi-channel data loggers, industrial PCs, motor control, robotics, industrial automation, smart sensors, portable instrumentation, and home medical appliances.

Operational Amplifiers

The MCP60X Operational Amplifier family includes four devices: MCP601, MCP602, MCP603 and MCP604. These devices are Microchip's first 2.7 volt single supply operational amplifier products. The MCP60X family offers a gain bandwidth product of 2.8 MHz with low typical operating current of 230 μ A. The MCP60X devices use Microchip's advanced CMOS technology which provides low bias current, high speed operation, high open-loop gain and rail-to-rail output swing.

System Supervisors

Microchip offers a complete family of system supervisor products. The new devices include the MCP809/810 and MCP100/101 supervisory circuits with push-pull output and the MCP120/130 supervisory circuits with open drain output. The devices are functionally and pin-out comparable to products from other analog suppliers.

Controller Area Network (CAN)

Microchip is enhancing its product portfolio by introducing the CAN Product Family. The MCP2510 is the smallest, easiest-to-use, CAN controller on the market today. Combining the MCP2510 with Microchip's broad range of high-performance PICmicro MCUs enables Microchip to support for virtually all of today's CAN-based applications. Other potential benefits of having a separate CAN controller include the ability for system designers to select from a much wider variety of MCUs for an optimal performance solution.

Additional products planned for Microchip's CAN product portfolio include other CAN peripherals and a family of PICmicro MCUs with integrated CAN support.

microID™ RFID Tagging Devices

Only Microchip manufactures world-class components for every application in the radio frequency identification (RFID) system. From the advanced, feature-packed microID family of RFID tags and high-endurance Serial EEPROMs to high performance PICmicro MCUs and KEELOQ code hopping encoders - Microchip's full range of RFID solutions are available for your tag, peripheral and reader application designs.

The microID family can emulate almost any standard on the market today. It provides drop-in compatible solutions to the most commonly used 125 kHz and 13.56 MHz tags and an upgrade migration path for virtually any application with higher performance and new features.

Serial EEPROM Overview

Microchip's high-endurance Serial EEPROMs complement the diverse MCU product families. Serial EEPROMs are available in a variety of densities, operating voltages, bus interface protocols, operating temperature ranges and space-saving packages.

Densities:

The densities range from 128 bits to 256 Kbits with higher density devices in development.

Bus Interface Protocols:

We offer all popular protocols: I²C™, Microwire® and SPI.

Operating Voltages:

In addition to standard 5V devices there are two low voltage families. The "LC" devices operate down to 2.5V, while the breakthrough "AA" family operates, in both read and write mode, down to 1.8V, making these devices highly suitable for alkaline and NiCd battery powered applications.

Temperature Ranges:

Like all Microchip devices, many Serial EEPROMs are offered in Commercial (0°C to +70°C), Industrial (-40°C to +85°C) and Extended (-40°C to +125°C) operating temperature ranges.

Packages:

Small footprint packages include: industry standard 5-lead SOT-23, 8-lead DIP, 8-lead SOIC in JEDEC and EIAJ body widths, and 14-lead SOIC. The SOIC comes in two body widths; 150 mil and 207 mil.

Technology Leadership:

Selected Microchip Serial EEPROMs are backed by a 1 million Erase/Write cycle. Microchip's erase/write cycle endurance is among the best in the world, and only Microchip offers such unique and powerful development tools as the Total Endurance disk. This mathematical software model is an innovative tool used by system designers to optimize Serial EEPROM performance and reliability within the application.

Microchip offers Plug-and-Play to the DIMM module market with the 24LCS52, a special function single-chip EEPROM that is available in space saving packages. For Plug-and-Play video monitor applications, Microchip offers the 24LC21, a single-chip DDC1™/DDC2™-compatible solution. In addition, Microchip released a high-speed 1 MHz 2-wire Serial EEPROM device ideal for high-performance embedded systems.

Microchip is a high-volume supplier of Serial EEPROMs to all the major markets worldwide. The Company continues to develop new Serial EEPROM solutions for embedded control applications.

Microchip Technology Inc.

include computer peripherals, instrumentation, and automotive devices. Microchip's expertise in surface mount packaging on SOIC and TSOP packages led to the development of the surface mount OTP EPROM market where Microchip is a leading supplier today. Microchip is also a leading supplier of low-voltage EPROMs for battery powered applications.

MIGRATABLE MEMORY™ TECHNOLOGY

Microchip's innovative Migratable Memory technology (MMT) provides socket and software compatibility among all of its equivalent ROM, OTP and FLASH memory MCUs. MMT allows customers to match the selection of MCU memory technology to the product life cycle of their application, providing an easy migration path to a lower cost solution whenever appropriate.

FLASH memory is an ideal solution for engineers designing products for embedded systems – especially during the development and early stages of the product. In certain products and applications, FLASH memory may be used for the life of the product because of the advantages of field upgradability or where product inventory flexibility is required.

Once the design enters the pre-production stage and continues through introduction and growth stages, OTP program memory provides maximum programming flexibility and minimum inventory scrappage. The OTP device is pin and socket compatible with the FLASH device – providing a lower cost, high-volume flexible solution.

As the design enters a mature stage and program code stabilizes, a lower cost, socket compatible ROM memory device could be used. In some cases, OTP memory may still be used as the most cost-effective memory technology for the product. Compatibility and flexibility are key to the success of the PICmicro MCU product family, and ultimately the success of our customers.

FLEXIBLE PROGRAMMING OPTIONS

To meet the stringent design requirements placed on our customers, the following innovative programming options are offered. These programming options address procurement issues by reducing and limiting work-in-process liability and facilitating finished goods code revisions. Microchip's worldwide distributors stock reprogrammable and one-time programmable inventory, allowing customers to respond to immediate sales opportunities or accommodate engineering changes off the shelf.

Reprogramming of the MCU program memory. Reprogrammability offers a highly flexible solution to today's ever-changing market demands – and can substantially reduce time to market. Users can program their systems very late in the manufacturing process or update systems in the field. This allows easy code revisions, system parameterization or customer-specific options with no scrappage. Reprogrammability also reduces the design verification cycle.

One-Time Programmable (OTP)

PICmicro OTP MCUs are manufactured in high volumes without customer specific software and can be shipped immediately for custom programming. This is useful for customers who need rapid time to market and flexibility for frequent software updates.

In-Circuit Serial Programming™ (ICSP™)

Microchip's PICmicro FLASH and OTP MCUs feature ICSP capability. ICSP allows the MCU to be programmed after being placed in a circuit board, offering tremendous flexibility, reduced development time, increased manufacturing efficiency and improved time to market. This popular technology also enables reduced cost of field upgrades, system calibration during manufacturing, the addition of unique identification codes to the system and system calibration. Requiring only two I/O pins for most devices, Microchip offers the most non-intrusive programming methodology in the industry.

Self Programming

Microchip's PIC16F87X family features self programming capability. Self programming enables remote upgrades to the FLASH program memory and the end equipment through a variety of medium ranging from Internet and Modem to RF and Infrared. To setup for self programming, the designer programs a simple boot loader algorithm in a code protected area of the FLASH program memory. Through the selected medium, a secure command allows entry into the PIC16F87X MCU through the USART, I²C or SPI serial communication ports. The boot loader is then enabled to reprogram the PIC16F87X FLASH program memory with data received over the desired medium. And, of course, self programming is accomplished without the need for external components and without limitations on the PIC16F87X's operating speed or voltage.

Quick-Turn Programming (QTP)

Microchip offers a QTP programming service for factory production orders. This service is ideal for customers who choose not to program a medium to high unit volume in their own factories, and whose production code patterns have stabilized.

Serialized Quick-Turn Programming (SQTPSM)

SQTP is a unique, flexible programming option that allows Microchip to program serialized, random or pseudo-random numbers into each device. Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

Masked ROM

Microchip offers Masked ROM versions of many of its most popular PICmicro MCUs, giving customers the lowest cost option for high volume products with stable firmware.

Future Products and Technology

Microchip is constantly developing advanced process technology modules and new products that utilize our advanced manufacturing capabilities. Current production technology utilizes lithography dimensions down to 0.7 micron.

Microchip's research and development activities include exploring new process technologies and products that have industry leadership potential. Particular emphasis is placed on products that can be put to work in high-performance broad-based markets.

Equipment is continually updated to bring the most sophisticated process, CAD and testing tools online. Cycle times for new technology development are continuously reduced by using in-house mask generation, a high-speed pilot line within the manufacturing facility and continuously improving methodologies.

Objective specifications for new products are developed by listening to our customers and by close co-operation with our many customer-partners worldwide.

Microchip Technology Inc.

NOTES:

SECTION 2 APPLICATION NOTES

Smart Sensor CAN Node using the MCP2510 and PIC16F876 - AN212	2-1
Temperature Sensing Technologies - AN679	2-83
Using Single Supply Operational Amplifiers in Embedded Systems - AN682.....	2-93
Single Supply Temperature Sensing with Thermocouples - AN684.....	2-101
Thermistors in Single Supply Temperature Sensing Circuits - AN685.....	2-117
Understanding and Using Supervisory Circuits - AN686.....	2-127
Precision Temperature Sensing with RTD Circuits - AN687	2-131
Layout Tips for 12-Bit A/D Converter Application - AN688.....	2-135
Understanding A/D Converter Performance Specifications - AN693	2-141
Interfacing Pressure Sensors to Microchip's Analog Peripherals - AN695.....	2-151
Anti-Aliasing, Analog Filters for Data Acquisition Systems - AN699	2-161
Interfacing Microchip MCP3201 A/D Converter to 8051-Based Microcontroller - AN702	2-171
Using the MCP320X 12-Bit Serial A/D Converter with Microchip PICmicro® Devices - AN703	2-183
Interfacing Microchip's MCP3201 Analog/Digital (A/D) Converter to MC68HC11E9-Based Microcontroller - AN704.....	2-205
Controller Area Network (CAN) Basics - AN713	2-215
Building a 10-bit Bridge Sensing Circuit using the PIC16C6XX and MCP601 Operational Amplifier - AN717	2-223
Interfacing Microchip's MCP3201 Analog-to-Digital Converter to the PICmicro® Microcontroller - AN719.....	2-231
Operational Amplifier Topologies and DC Specifications - AN722.....	2-251
Operational Amplifier AC Specifications and Applications - AN723	2-261
Using the MCP2510 CAN Developer's Kit - AN733	2-271



MICROCHIP



MICROCHIP

AN212

Smart Sensor CAN Node using the MCP2510 and PIC16F876

Author: Mike Stanczyk
Diversified Engineering

INTRODUCTION

Advances in data communications has created efficient methods for several devices to communicate over a minimum number of system wires. The Controller Area Network (CAN) is one of these methods. CAN sends and receives messages over a two wire (CAN) bus. The nodes broadcast their individual messages over the CAN bus, while the receivers that are set-up to accept the message and anticipate an acknowledgment signal indicating the receipt of a non-corrupted message. The protocol of the CAN has two states and the bits are either dominant (logic "0") or recessive (logic "1"). Nodes may attempt to transmit a message at the same time. To ensure that collisions do not reduce the throughput of the bus there is an arbitration scheme. In this scheme, a node will continue to transmit until a dominant bit is detected while that node is expecting a recessive bit on the bus (in the ID field). The node(s) that lost arbitration will automatically terminate their transmission and switch to receive mode. Some time later after the CAN bus enters an idle state, these nodes attempt to re-transmit. If the node did not lose arbitration, it completes its transmission. (For additional information on the CAN protocol, refer to AN713, "CAN Basics").

The bus configuration operates by the multi-master principle, and it allows several Node Boards to connect directly to the bus. If one Node Board fails in the system, the other Node Boards are not affected. The probability of the entire network failing is extremely low compared to ring type networks. Ring type networks have a high probability failure rate, due to the fact that if one node malfunctions, the entire network becomes inoperable. The CAN controller seeks to solve this problem.

MCP2510 CAN Controller Benefits

- Monitors Several Devices
- Individual Node Programming
- Replaces a Large Wiring Harness

SYSTEM OVERVIEW

The system hardware can be broken down into two components. These are:

- CAN-NET Node Board
- CAN-NET Analog Input Board

These boards can be purchased from Diversified Engineering in the form of their CAN-NET Analog Input Node Kit. The CAN-NET Analog Input board also requires that some of the customer installed options be installed. The two additional components are a 14.5-PSI Pressure Transducer and an LED. Table 1 gives the part numbers for two of the components.

TABLE 1: COMPONENT PART NUMBERS

Manufacturer	Component	Part Number
Diversified Engineering	CAN-NET Analog Input Node Kit	905190
Motorola	Pressure Transducer	MPX2010DP

This system has several key features. These include:

- High Speed SPI Interface
- MPLAB®-ICD Debugging Tool
- Low Power CMOS Technology
- PWM Output for Driving a Lamp
- Supports SPI modes 0,0 and 1,1

2

Application Notes

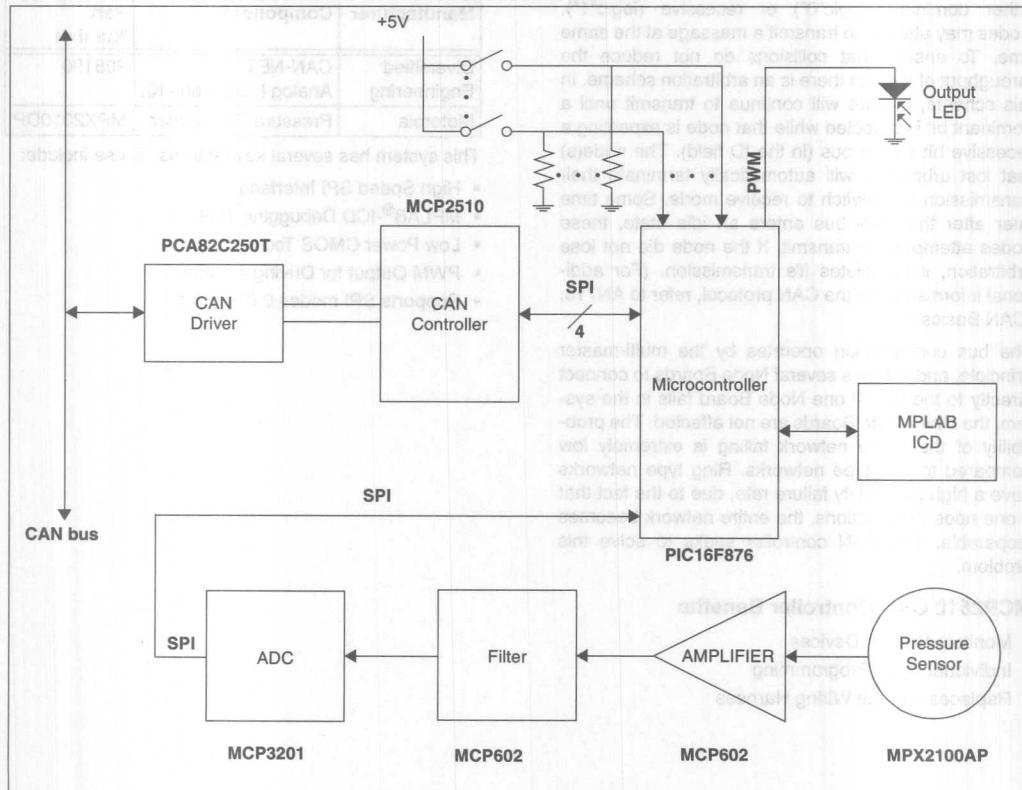
CAN-NET Node Board

The CAN-NET Node Board consists of hardware devices that are used in conjunction with software programming techniques to achieve an optimal Controller Area Network. The versatility of the CAN controller enables a wide variety of applications based on the concept of this particular design.

The MCP2510 CAN controller is the heart of the CAN interface. It handles all of the transmitting and receiving of message packets, that contain useful information for other nodes on the network via the CAN bus. The MCP2510 CAN controller is also designed to interface with the Serial Peripheral Interface (SPI) port. The SPI port is available on the PIC16F876 microcontroller, and the MCP3201 analog to digital converter.

The PIC16F876 microcontroller stores the program in memory and reads the DIP switch settings for sending and receiving messages. It controls the PWM output and enables the MPLAB® ICD to be used as a debugging tool.

FIGURE 1: BLOCK DIAGRAM OF THE CAN NODE BOARD



CAN-NET Analog Input Board

The MCP3201 analog to digital converter accepts input signals from the pressure sensor, utilizing a differential amplifier configuration. The MCP602 amplifier package uses single supply CMOS operational amplifier technology.

HARDWARE OVERVIEW

This section describes the CAN-NET Node board hardware and how the CAN functions in the Node Board system. Schematics can be found in Appendix A.

MCP2510 CAN Controller

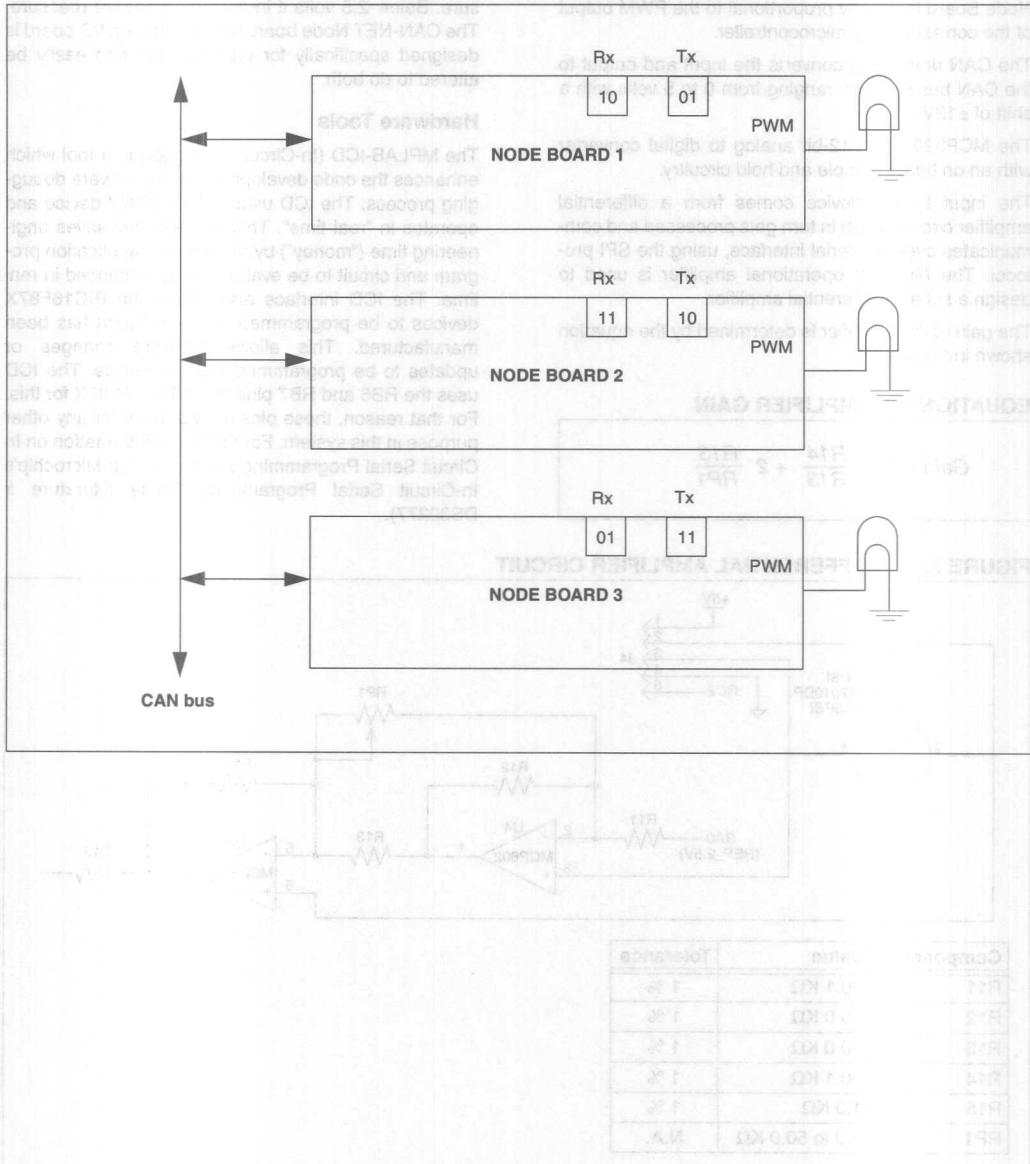
The high level design of this system is shown in Figure 1. The concept is to enable the MCP2510 CAN controller, the PIC16F876 microcontroller and the MCP3201 analog to digital (A/D) converter to efficiently communicate among each other utilizing the Serial Peripheral Interface (SPI). The MCP2510 handles the lower level protocols.

The PIC16F876 microcontroller stores the program in memory and constantly polls the MCP3201 analog to digital (A/D) converter, along with the reference A/D.

In the main loop of the program, a variable is toggled. When the value of the variable is a logic "0", the PICmicro® device reads the pressure sensor and when the value of the variable is a logic "1", the PICmicro device reads the reference A/D. The microcontroller also

reads the settings of the input switches. The first two (of four) switches tells the microcontroller which message the node is allowed to receive. The last two (of four) switches tells the microcontroller the transmit address of the node. The configuration shown in Figure 2 illustrates three Node Boards on a CAN bus and they are set to transmit and receive certain messages.

FIGURE 2: THREE NODE BOARDS CONNECTED TO THE CAN BUS



Node 1 is set to receive a value of the pressure sensor from a different node. The identification for each Node Board is 01, 10, and 11. These settings are both for transmit and receive identifiers. Node Board 1 is set to receive the pressure sensor value from Node Board 2, Node Board 2 is set to receive the pressure sensor value from Node Board 3, and Node Board 3 is set to receive the pressure sensor value from Node Board 1. The pressure sensor value of each Node Board is directly proportional to the PWM output of the corresponding microcontroller.

The CAN driver chip converts the input and output to the CAN bus voltages ranging from 0 to 5 volts with a shift of $\pm 12V$.

The MCP3201 is a 12-bit analog to digital converter with an on board sample and hold circuitry.

The input to the device comes from a differential amplifier circuit, which in turn gets processed and communicates over the serial interface, using the SPI protocol. The MCP602 operational amplifier is used to design a suitable differential amplifier.

The gain of the amplifier is determined by the equation shown in Equation 1.

EQUATION 1: AMPLIFIER GAIN

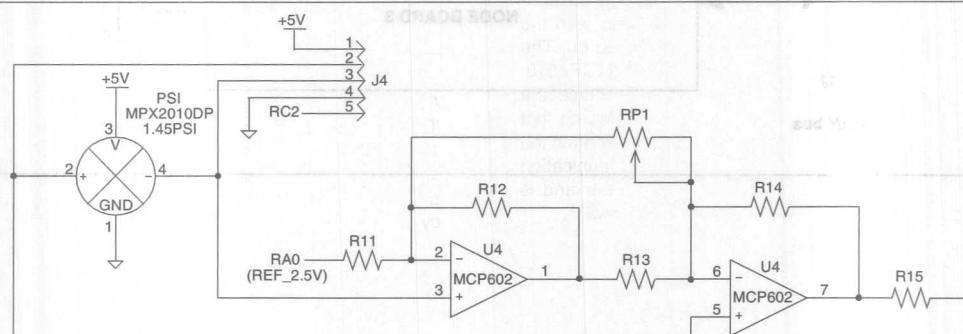
$$Gain = 1 + \frac{R14}{R13} + 2 \frac{R13}{RP1}$$

to this amplifier ranges from 0 to 5 volts and is useful in this application for pressure. The pressure can be referred to as zero pressure and the normalized pressure setting, negative pressure when things suck and positive pressure when things blow. The pressure sensor puts out a negative voltage when there is negative pressure and a positive voltage when there is positive pressure. The reference for the differential amplifier is 2.5 volts. Above 2.5 volts it indicates a positive pressure. Below 2.5 volts it indicates a negative pressure. The CAN-NET Node board with the Analog I/O board is designed specifically for pressure, but can easily be altered to do both.

Hardware Tools

The MPLAB-ICD (In-Circuit Debugger) is a tool which enhances the code development and hardware debugging process. The ICD uses a PIC16F877 device and operates in "real time". This low cost tool saves engineering time ("money") by allowing the application program and circuit to be evaluated and enhanced in real time. The ICD interface also allows the PIC16F87X devices to be programmed after the board has been manufactured. This allows software changes or updates to be programmed into the device. The ICD uses the RB6 and RB7 pins of the PIC16F87X for this. For that reason, these pins are not used for any other purpose in this system. For additional information on In-Circuit Serial Programming please refer to Microchip's In-Circuit Serial Programming Guide (literature # DS30277).

FIGURE 3: DIFFERENTIAL AMPLIFIER CIRCUIT



Component	Value	Tolerance
R11	30.1 KΩ	1 %
R12	10.0 KΩ	1 %
R13	10.0 KΩ	1 %
R14	30.1 KΩ	1 %
R15	1.0 KΩ	1 %
RP1	0.0 to 50.0 KΩ	N.A.

SOFTWARE OVERVIEW

Programming Style

The code for the Node Board is written in the PICmicro device instruction set to be assembled using Microchip's MPLAB environment. There is a significant use of macros to make the code more readable and less error prone. The macros are defined in three files:

1. Near the top of the main file,
2. CANLIB.ASM (file contains the CAN macros)
3. MACROS16.INC

If an unfamiliar instruction is found, it is probably made up of a set of familiar instructions in one of the macros. The macros in MACROS16.INC file are used extensively in writing code for the PICmicro microcontroller family, because they increase readability and greatly reduce programming errors.

Common Code

The Node Board uses common software files to maximize the program's efficiency. The routines that provide communication with the MCP2510 CAN chip are in the file CANLI.ASM and the definitions of the MCP2510 registers are in MCP2510.INC. The common macros are in MACROS16.INC.

SPI Communications

Communications from a device on the node (such as a microcontroller) to the MCP2510 are through the Serial Peripheral Interface (SPI) bus. The PICmicro device used on the Node Board fully supports the SPI in the master mode. Command strings are sent and received using a single software buffer. To send a string, the software buffer, called `pSPIBufBase`, is loaded with the bytes to send and the SPI interrupt is turned on. The interrupt handler exchanges bytes with the MCP2510. The bytes received from the MCP2510 replace the bytes that were sent from the software buffer, so that after the string has been sent, the buffer will contain the bytes received from the MCP2510. All communication with the MCP2510 is handled in this manner and is encapsulated in the routines in the CANLIB.ASM.

General ID Structure

The ID structure used by the Node Boards is determined by the settings on the DIP switches on power-up or after a reset. Changing the DIP switches while running has no effect on the ID structure.

Receive ID Structure

The Node Board uses the following setting in Table 2 for receiving:

TABLE 2: RECEIVING SETTINGS FOR THE NODE BOARD

Register	Value
RxMask0	0xFF
RxMask1	0xFFE
RxFilter0	0xFF
RxFilter1	0xFF
RxFilter2	0x00 ⁽¹⁾
RxFilter3	0x10
RxFilter4	0xFF
RxFilter5	0xFF

Note 1: This value is the Base Receive ID for receiving. The DIP #1 and DIP #2 settings are used to determine this value.

The DIP settings for receiving are shown in Table 3.

TABLE 3: DIP SWITCH ID SETTINGS FOR RECEIVING

DIP #1	DIP #2	ID
0	0	0x000
0	1	0x100
1	0	0x200
1	1	0x300

A message received for RxFilter2, i.e. Base Receive ID, is assumed to be a two-byte integer that contains a 12-bit value, i.e. between 0 and 4095. The 12-bit data is used to generate a PWM output where a 0 generates a 0% duty cycle and 0xFFFF generates a 100% duty cycle.

Transmit ID Structure

The Node Board transmits a CAN message every 131ms. A message contains two data bytes that represent a 12-bit value with least significant byte sent first.

The pressure switch is assigned to the Base Transmit ID and is measured and transmitted with that ID every 393ms, as a two-byte integer in the range 0 to 4095. Note that the A/D measurement is 8 bits that is then shifted by 4 bits before transmission, hence, its actual range is 0x0000 to 0xFF00.

Each data source has its own unique ID obtained from the Base Transmit ID, that is obtained from DIP #3 and DIP #4 and is shown in Table 4.

TABLE 4: DIP SWITCH ID SETTINGS FOR TRANSMITTING

DIP #3	DIP #4	ID
0	0	All transmissions are disabled
0	1	0x100
1	0	0x200
1	1	0x300

TABLE 5: SOFTWARE FUNCTION DESCRIPTIONS

Function Name	Function Description	Figure Number
Main	This is the main loop of the program	Figure 4
Hardstart	Do a full initialization of the system	Figure 5
Init	Initialize the PIC16F87X registers	Figure 6
InitSPIPort	Initialize the PIC16F87X SPI port	Figure 7
Init2510	Initialize the MCP2510's registers	Figure 8
Read3201	Read the specified register in the MCP3201 (A/D converter)	Figure 9
ReadA2D	Read the specified register in the MCP3201 (A/D converter)	Figure 10
WaitANDeqZ	Wait for pending messages	Figure 11
CheckCANMsg	Checks for messages in the receive buffer	Figure 12
ParseCAN	Setup the message for the PWM output	Figure 13
Reset2510	Resets the MCP2510	Figure 14
BitMod2510	Modifies the value of a specified bit in the MCP2510	Figure 15
Wrt2510Reg	Write the specified register in the MCP2510 (CAN interface)	Figure 16
SetNormalMode	Sets the MCP2510 to normal operating mode	Figure 17
Rd2510Reg	Read the specified register in the MCP2510 (CAN interface)	Figure 18
OutputPWM	Loads the PWM duty cycle registers with the values in the specified registers	Figure 19
InitSPIBuf	Initializes SPI buffer for transaction	Figure 20
LoadSPISByte	Loads the value in the W register into the SPI buffer	Figure 21
ExchangeSPI	Initiates the SPI transaction	Figure 22
WaitSPIExchange	Waits for the SPI transaction to be completed	Figure 23
LoadSPIZeros	Clears the value in the SPI buffer	Figure 24

The MCP2510 CAN controller has a 125k bit rate and the polling method is used. The use of interrupts would be easier in the system, but polling allowed the interrupt pins to remain free for other potential functions in the system.

There are three ways of transmitting information. The first way, finds out that something has actually happened and is basically triggered by an event. The second way, transmits the register status but it can not tell when an event happened. The third way, is the combination of the first two. This is an asynchronous method that schedules the message. If there happens to be a failure in transmission, the transmission thinks that it told the receiver, but in actuality it did not. It depends on the state of the device and that is the problem with event driven transmissions.

The flowcharts for the operation of the source code are shown in the Figure 4 through Figure 24. The subroutines contain the actual name and the function it performs within the flowchart, so that it can easily be referenced with the source code. Table 5 gives the function names used and a brief description of the function. In the electronic version of this document, clicking of the function name will take you (link you) to the page for that function.

CONCLUSION

The MCP2510 offers a simple method to interface a CAN network in order to maximize the transmitting and receiving of data via the CAN Bus. This efficient method allows a wide variety of I/O devices to be connected to the network using a Node Board. An advantage in utilizing a system of this nature is to be able to monitor several Node Boards at any given time. If an error occurs, it is detected and re-transmitted over the bus line until the receiver acknowledges the message. Another advantage is that several Node Boards can work off of one bus line rather than using a large wiring harness that connects to a main control panel. Our design demonstrated a useful way to implement a simple input pressure switch connected to a Node Board along with a visual light source to display the value in terms of brightness. By this example, several uses for different types of inputs and outputs can be implemented simply by using the basic techniques from this design.

Table 6 shows the resource requirements for the major functions.

TABLE 6: PIC16CXXX REQUIREMENTS

Function	Memory		Instruction Cycles
	Program	Data	
Assemble Message	TBD	TBD	TBD
Transmit	TBD	TBD	TBD
Receive	TBD	TBD	TBD
Disassemble Message	TBD	TBD	TBD
Test Error Conditions	TBD	TBD	TBD

CONTACTING DIVERSIFIED ENGINEERING

Additional information and CAN related products may be acquired from Diversified Engineering. You may contact them by either calling:

(203) 799-7875

or by visiting their web site:

www.DiversifiedEngineering.net

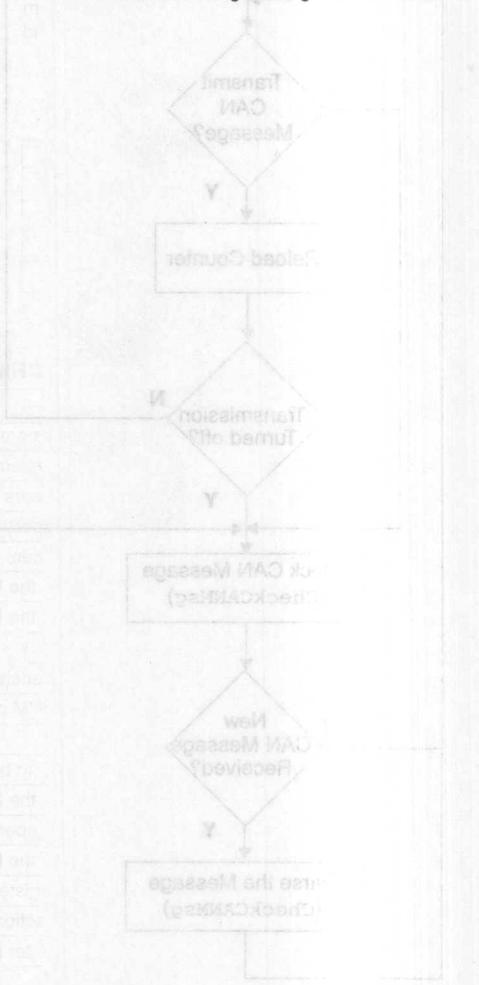


FIGURE 4: MAIN PROGRAM LOOP (Main)

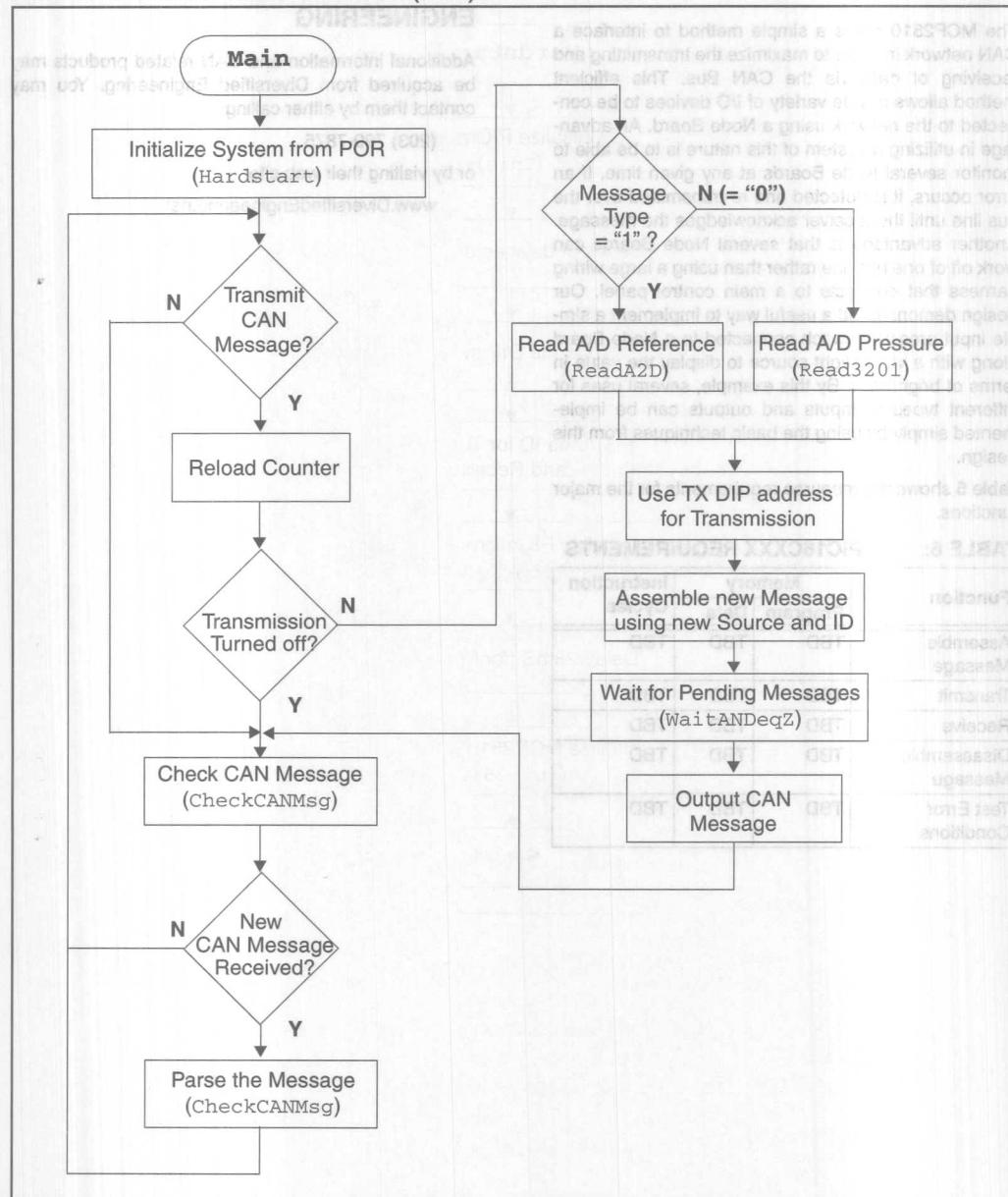


FIGURE 5: HARDSTART (Hardstart)

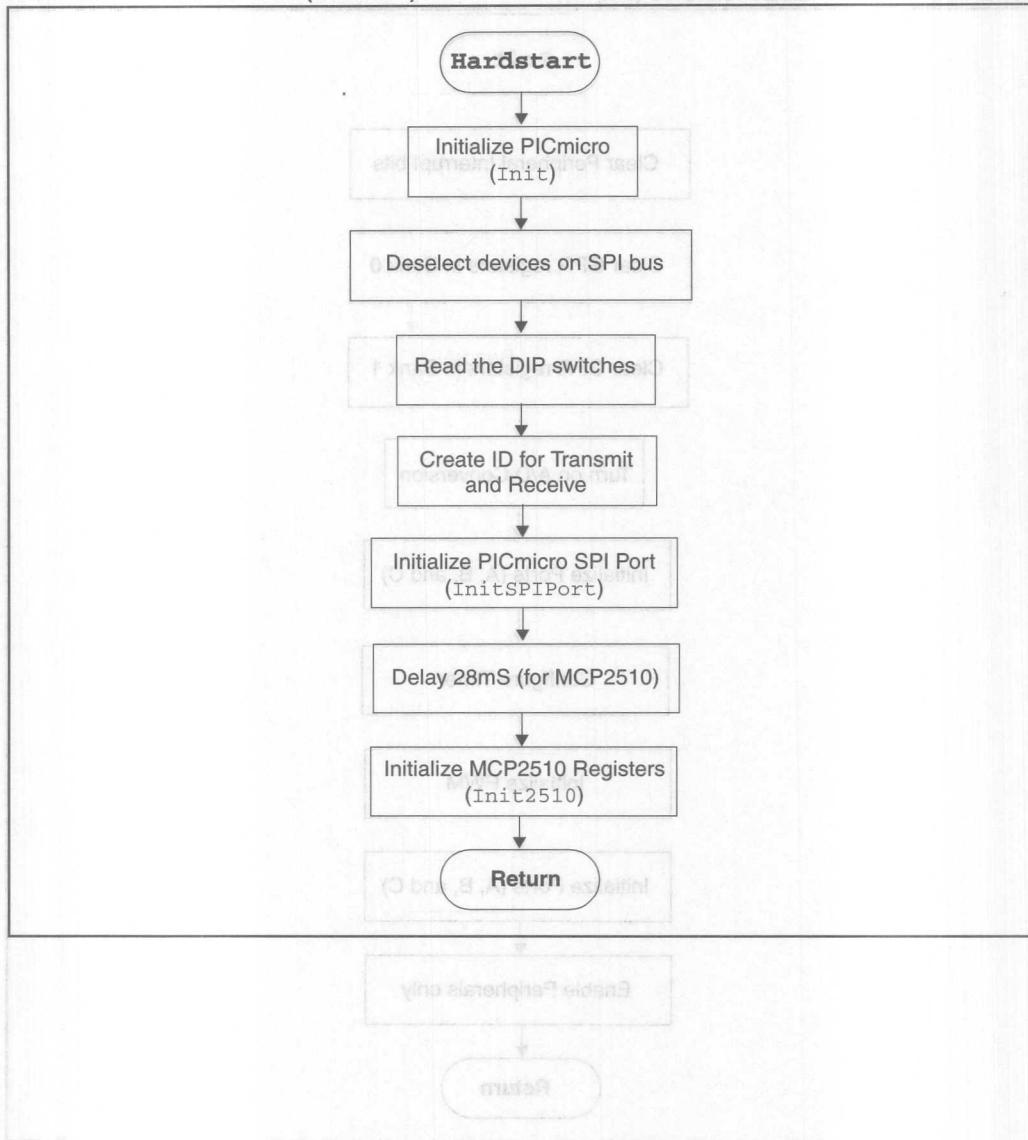


FIGURE 6: INITIALIZE PICMICRO (Init)

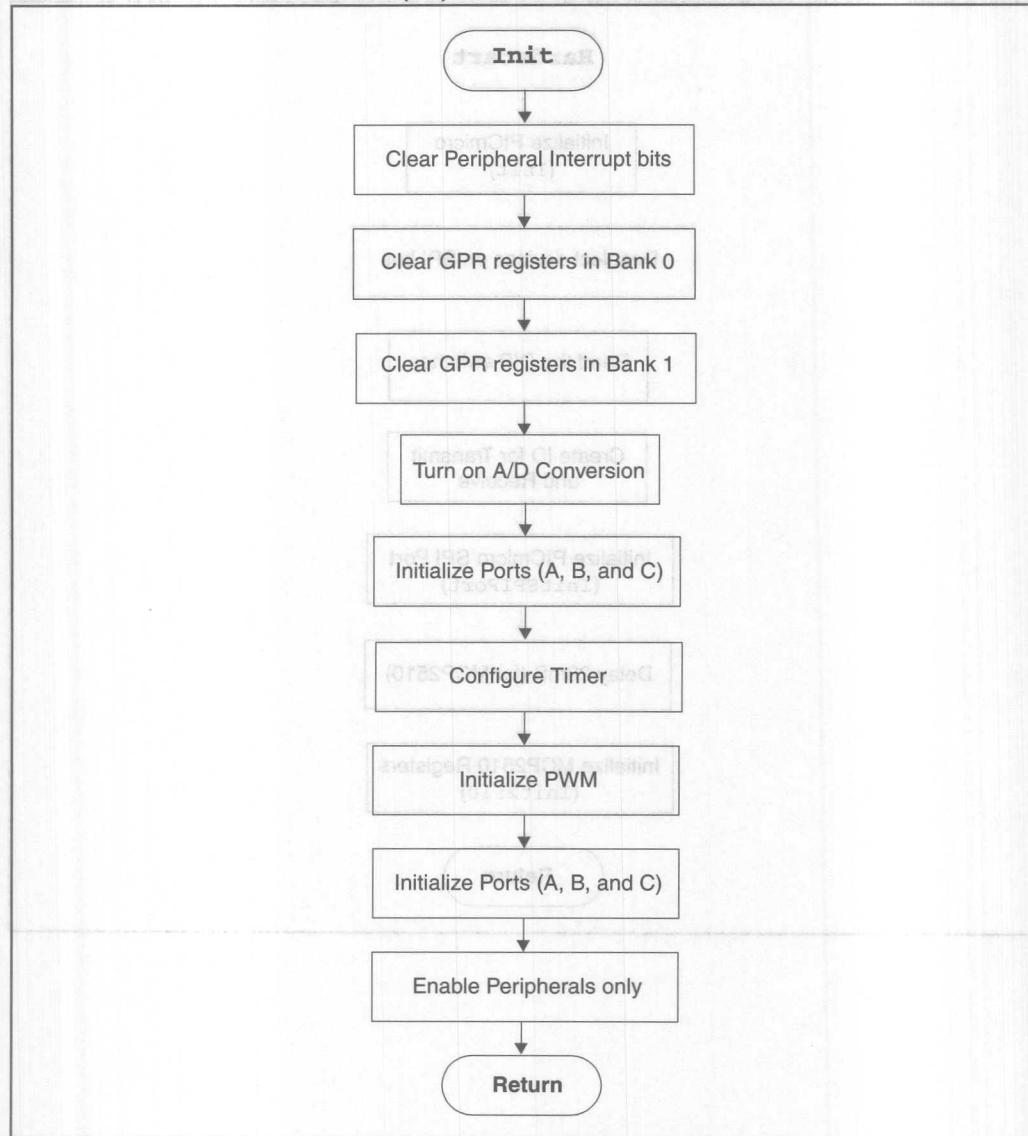


FIGURE 7: SETUP SPI PORT (InitSPIPort)

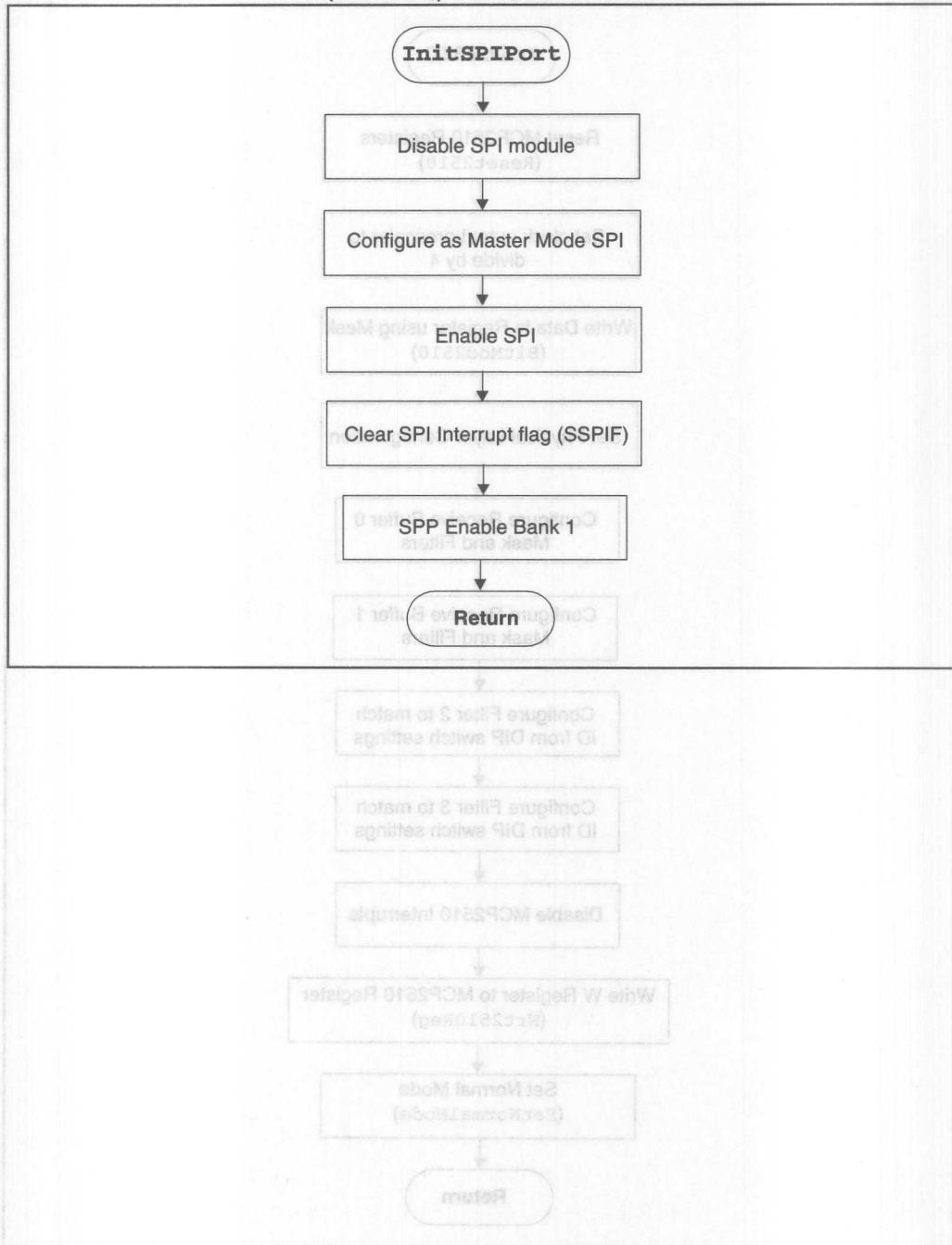


FIGURE 8: SETUP MCP2510 REGISTERS (Init2510)

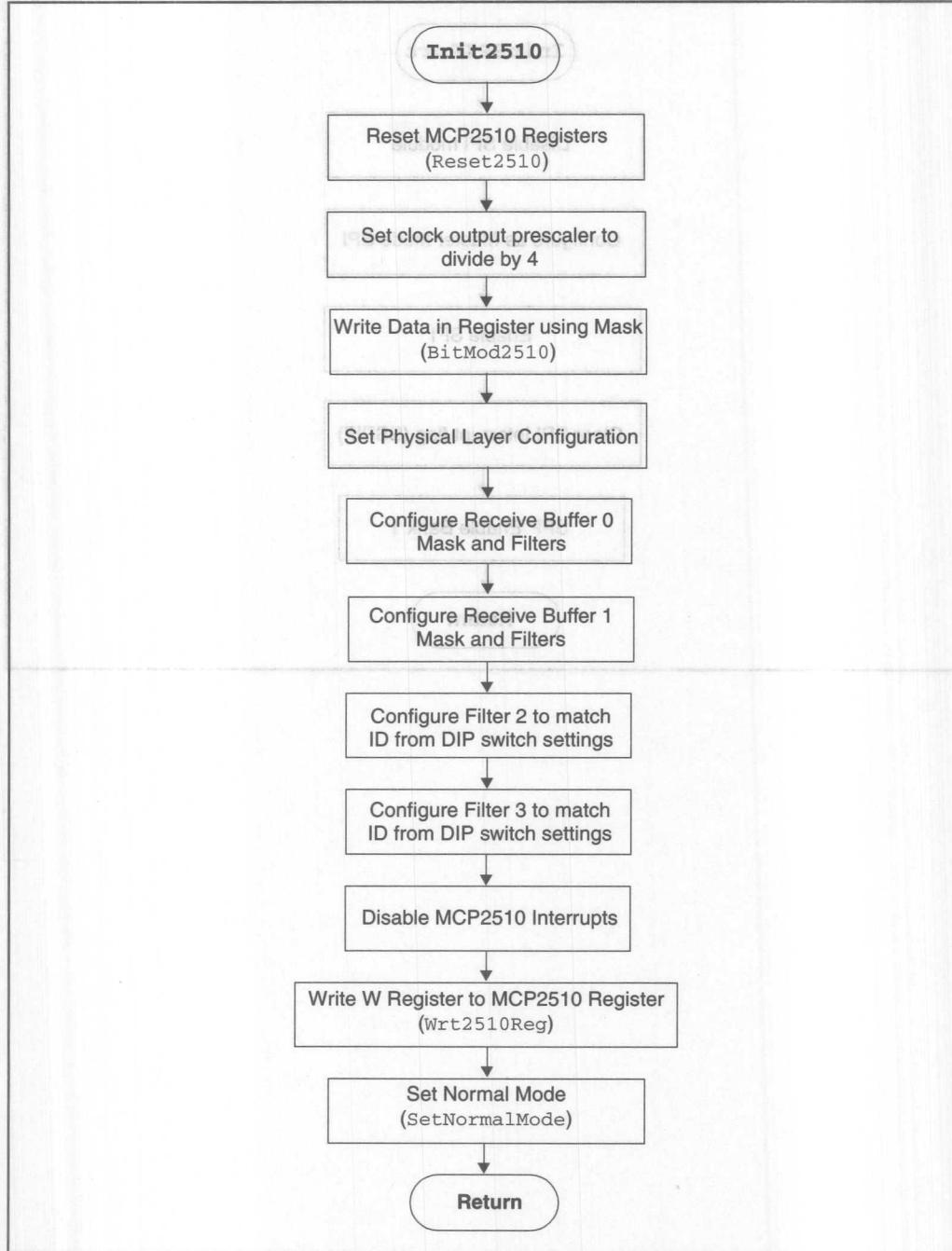


FIGURE 9: READ A/D PRESSURE (Read3201)

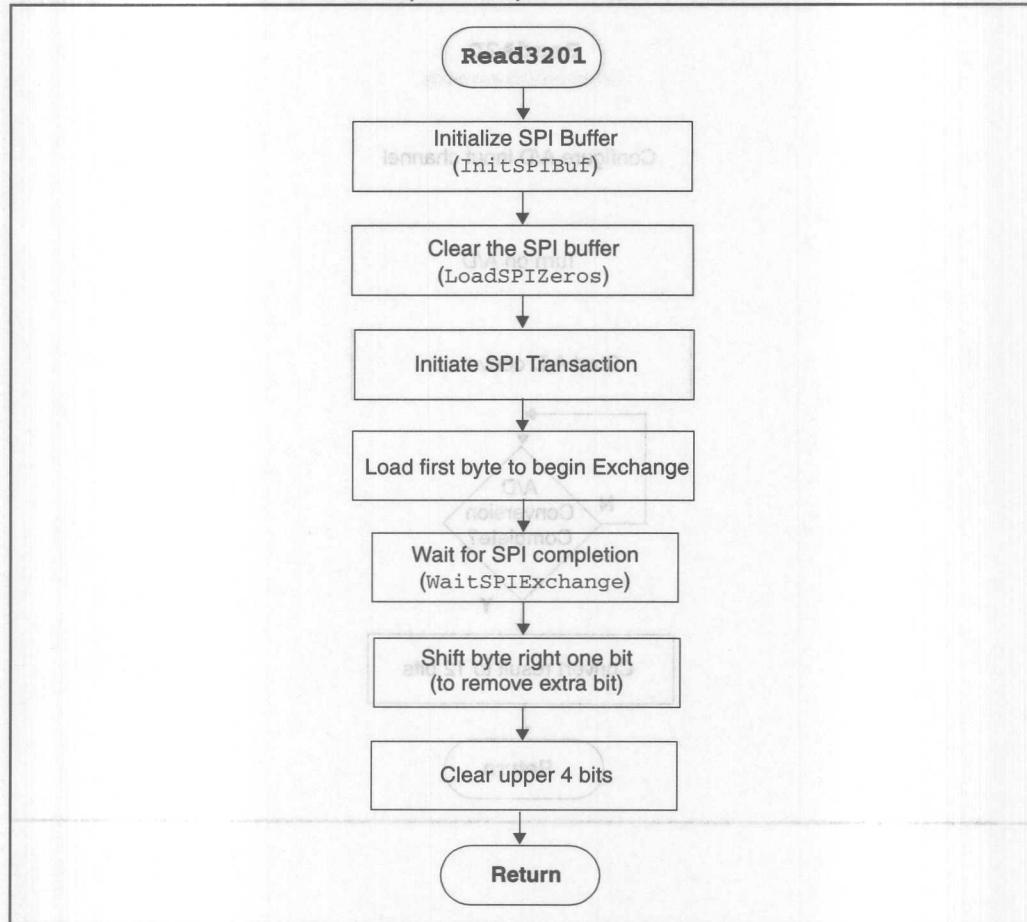


FIGURE 10: READ A/D REFERENCE (ReadA2D)

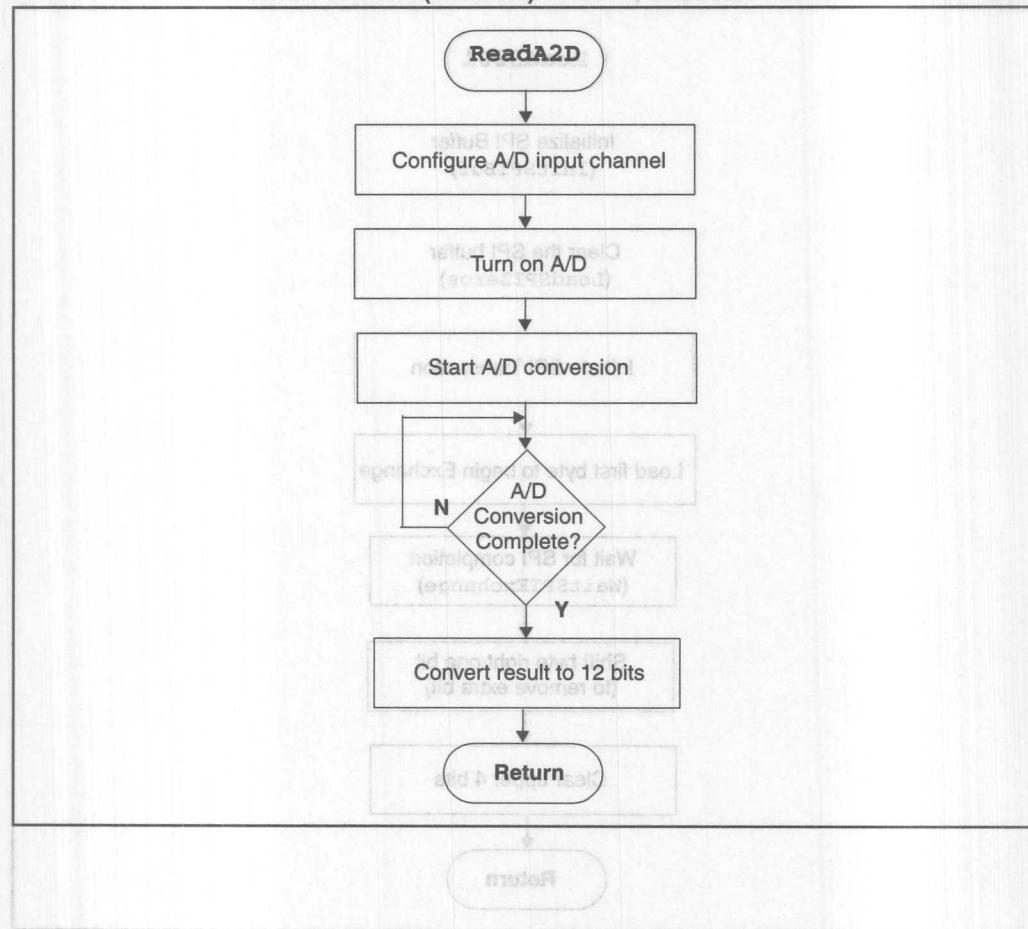
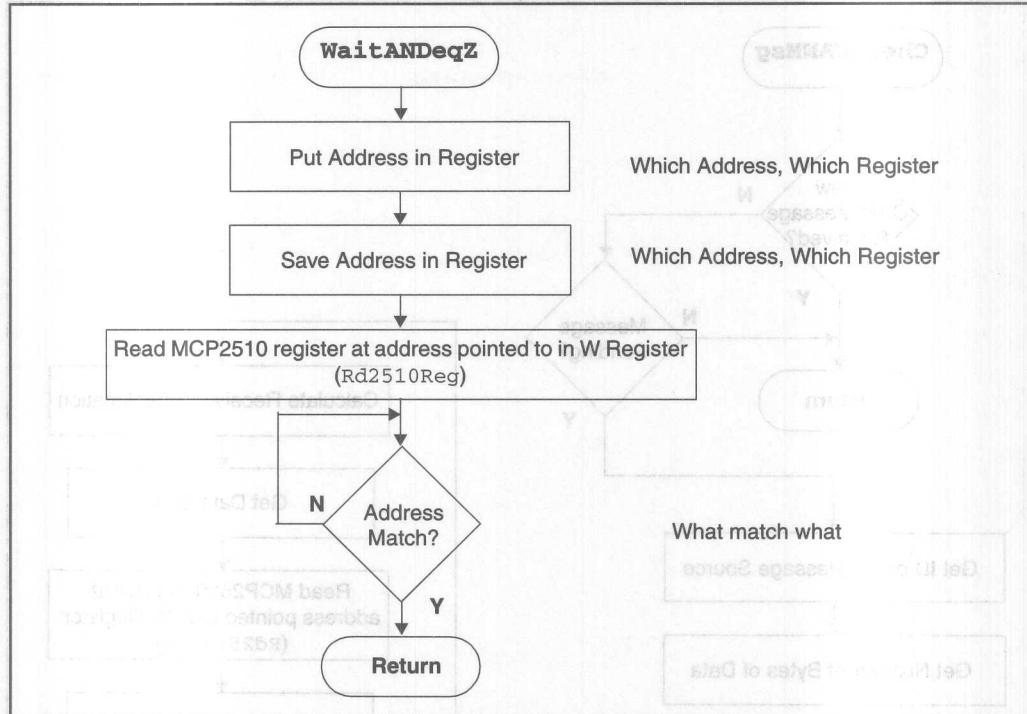


FIGURE 11: WAIT FOR PENDING MESSAGES (WaitANDeqZ)



AN212

FIGURE 12: CHECK CAN MESSAGE (CheckCANMsg)

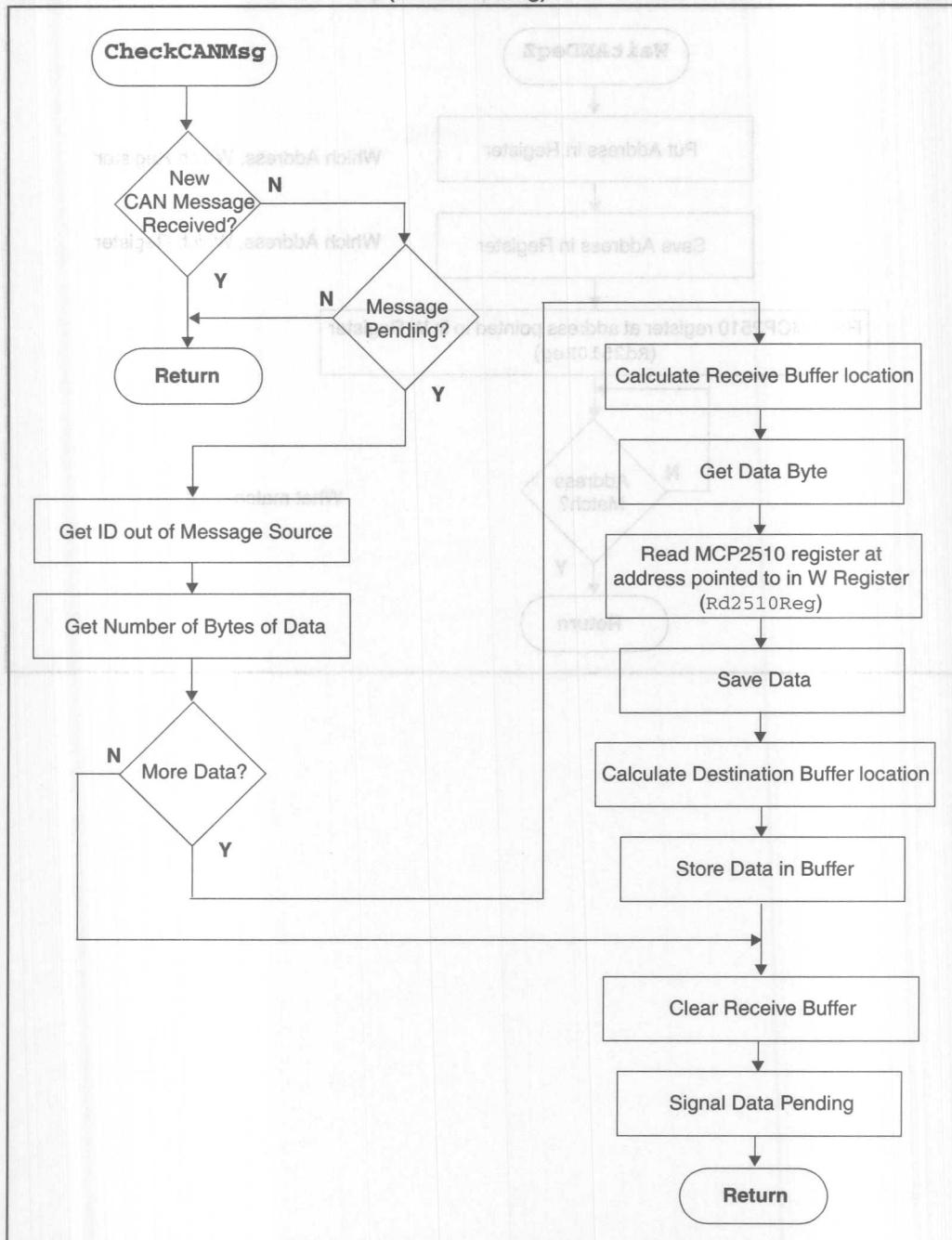


FIGURE 13: PARSE THE MESSAGE (ParseCAN)

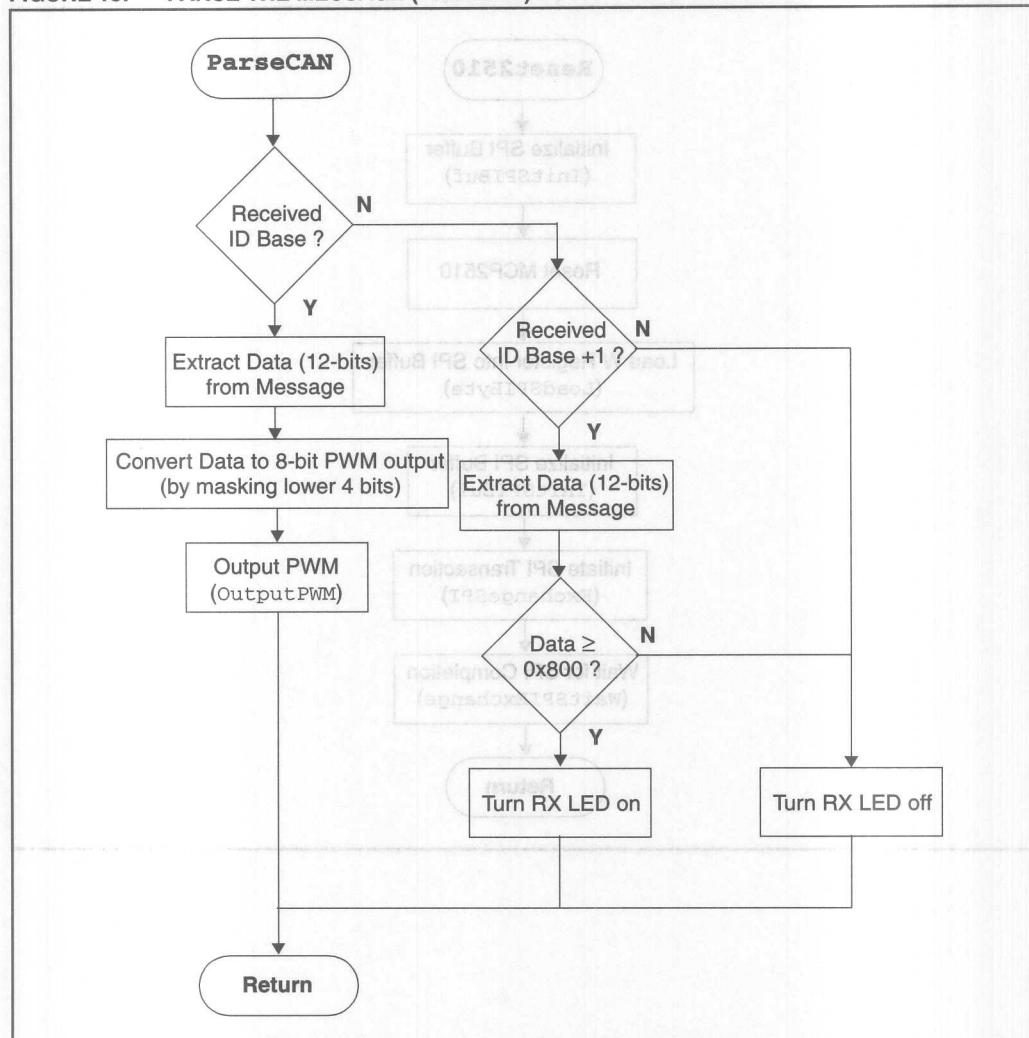


FIGURE 14: RESET MCP2510 REGISTERS (Reset2510)

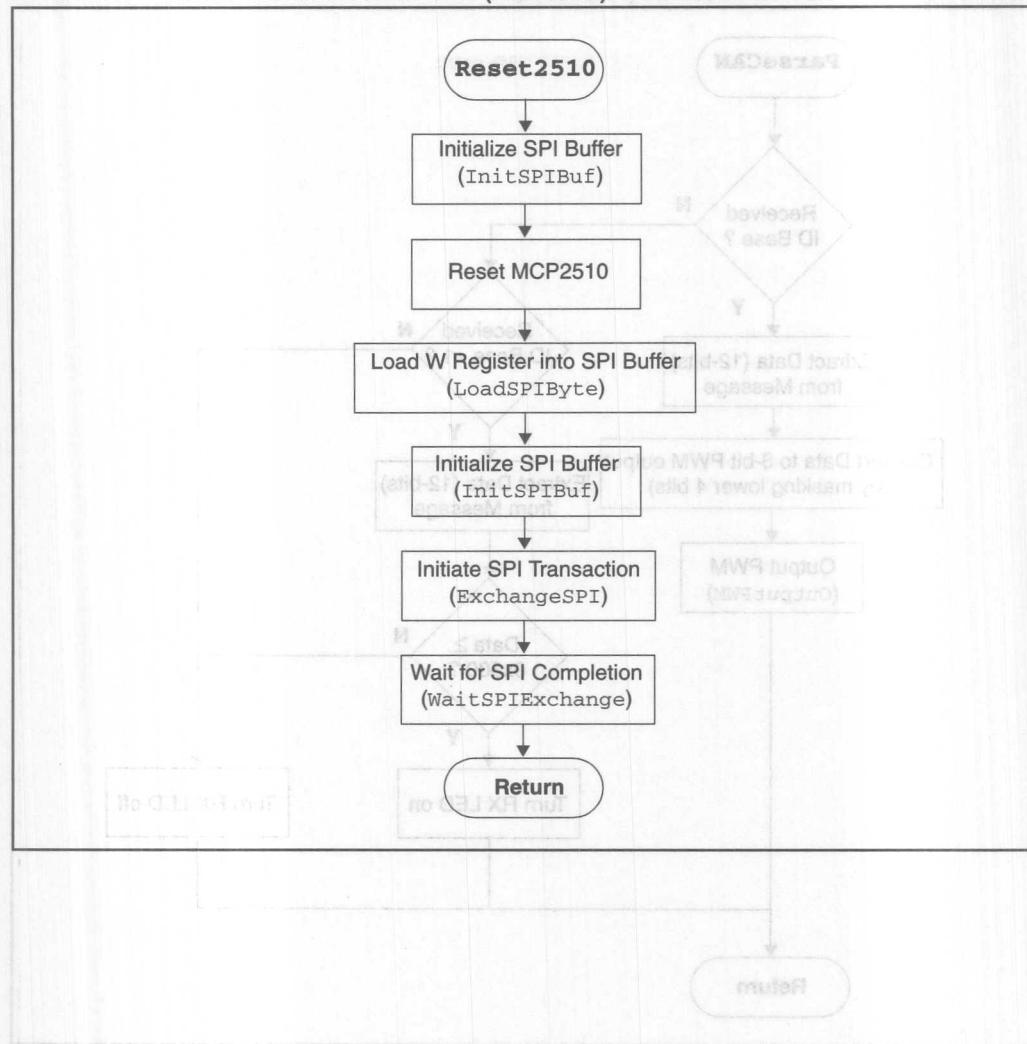


FIGURE 15: WRITE DATA IN REGISTER USING MASK (BitMod2510)

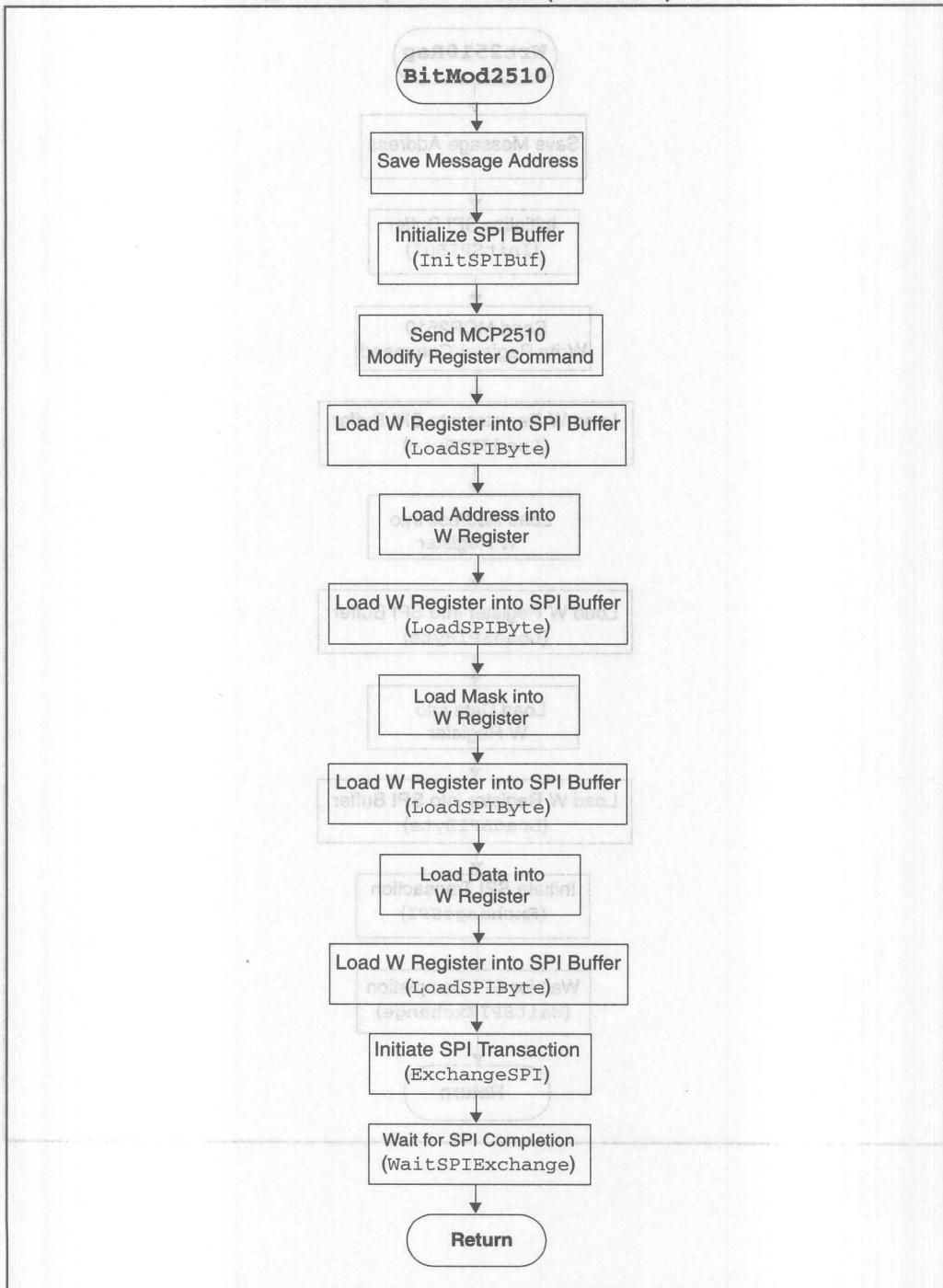


FIGURE 16: WRITE BYTE IN MCP2510 REGISTER IN W (Wrt2510Reg)

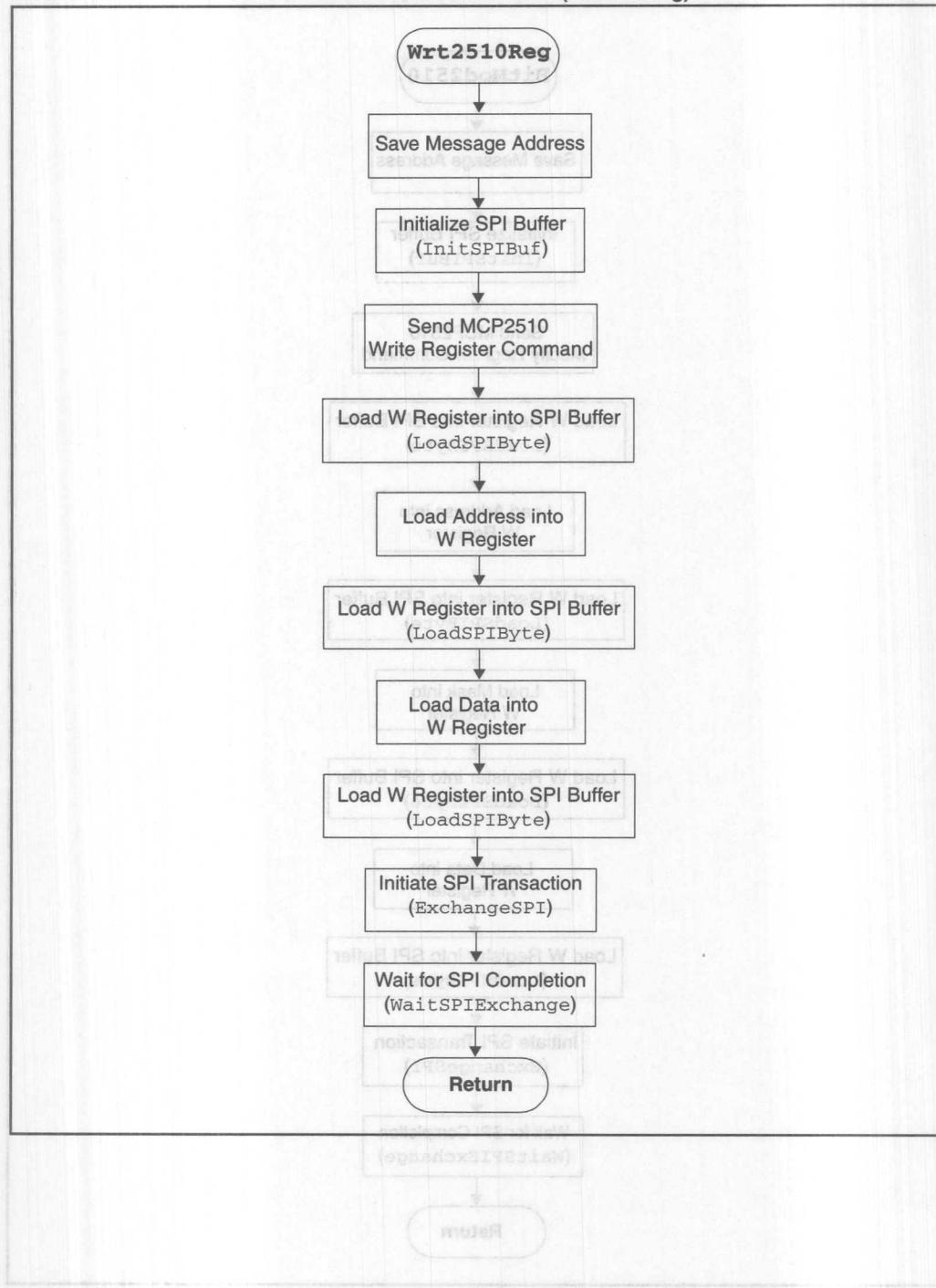


FIGURE 17: SET NORMAL MODE (SetNormalMode)

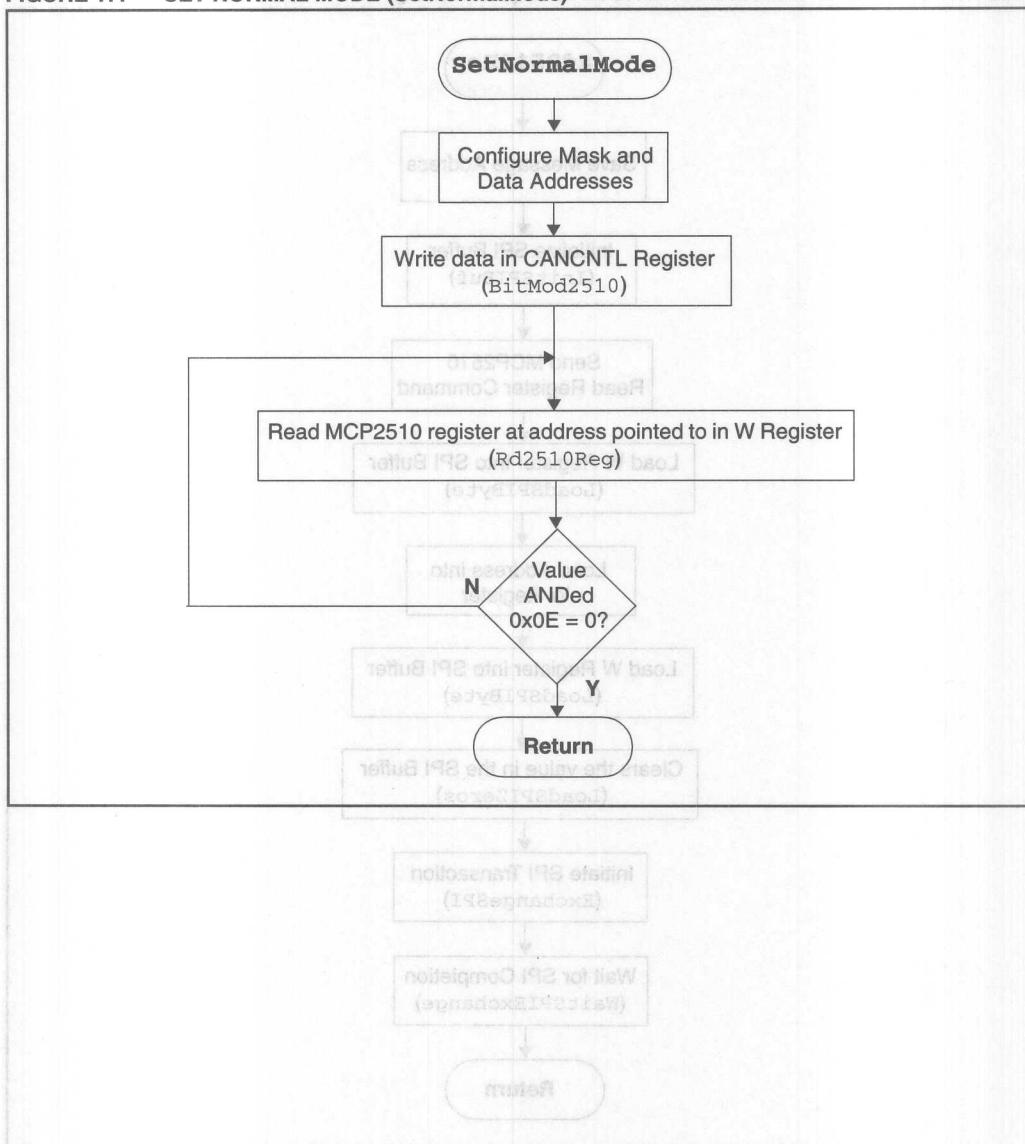


FIGURE 18: READ REGISTER ADDRESS IN W (Rd2510Reg)

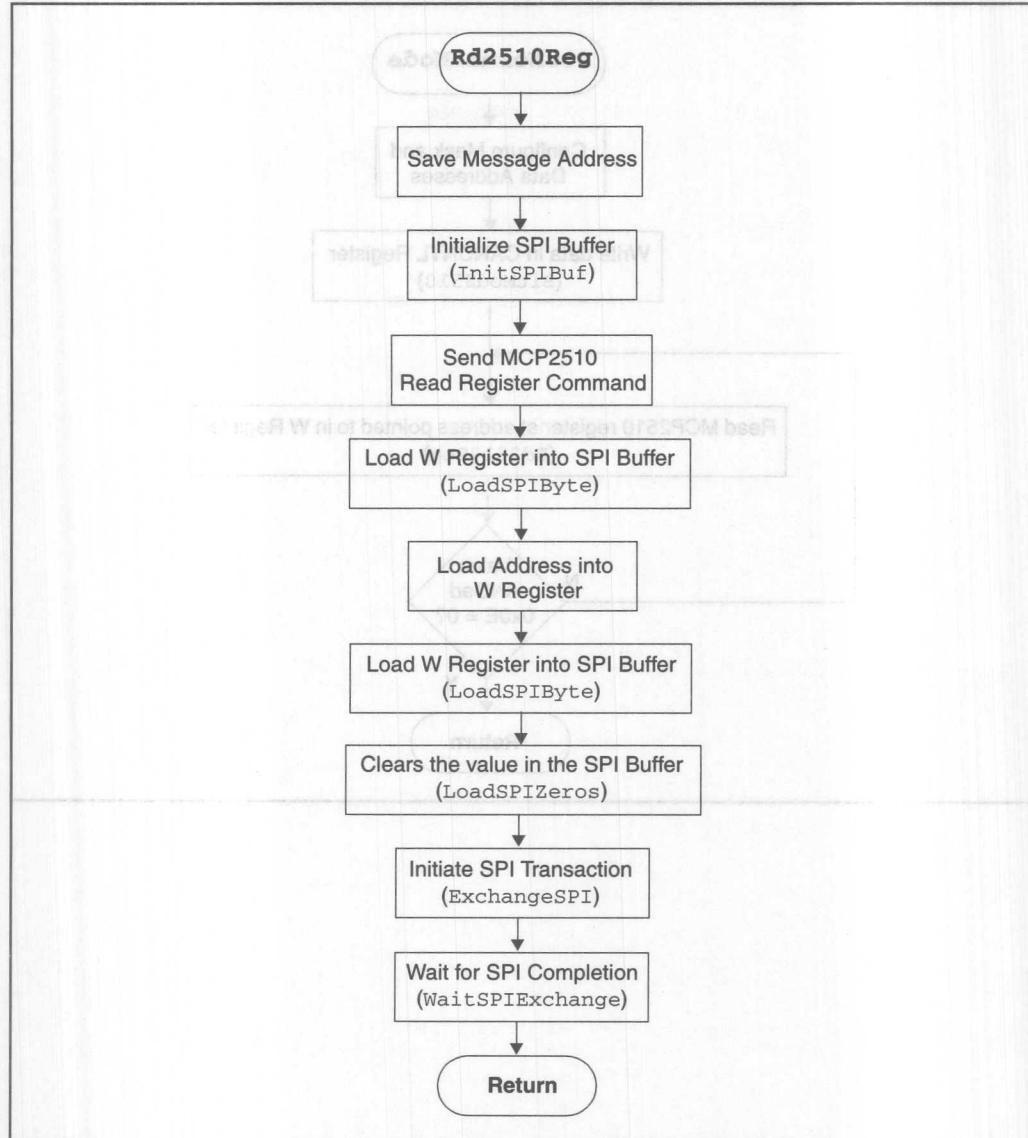


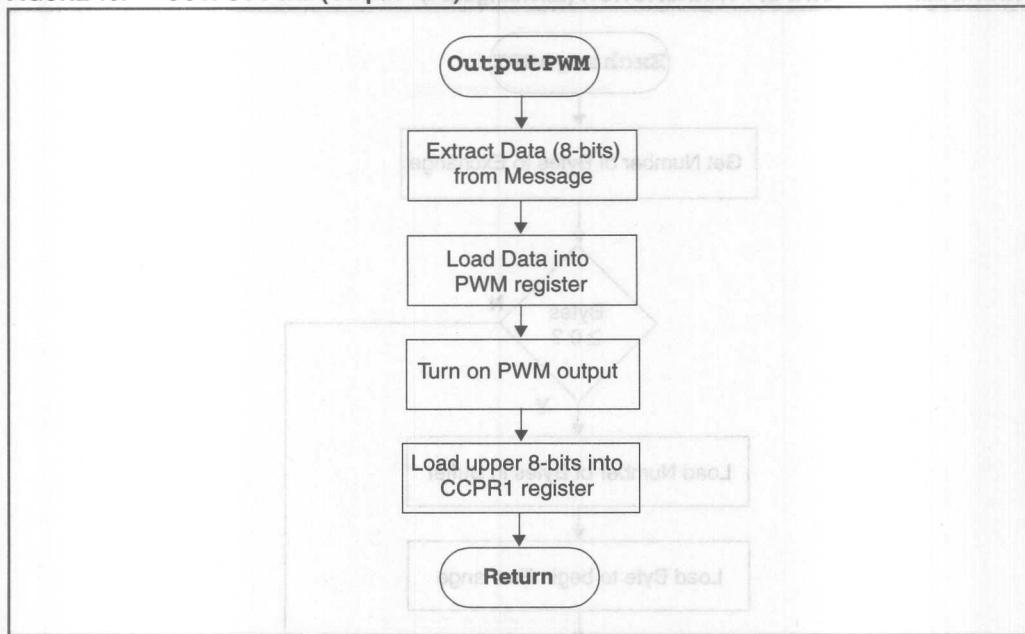
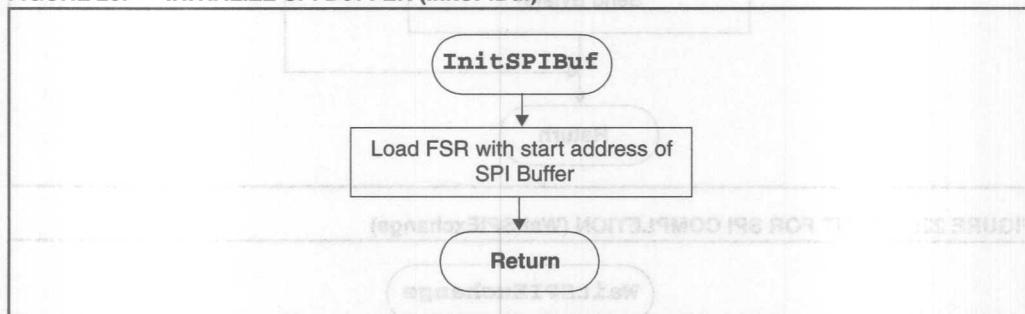
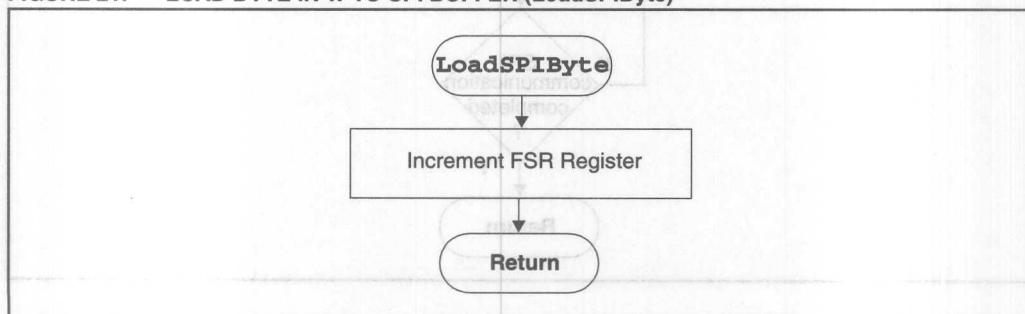
FIGURE 19: OUTPUT PWM (OutputPWM)**FIGURE 20: INITIALIZE SPI BUFFER (InitSPIBuf)****FIGURE 21: LOAD BYTE IN W TO SPI BUFFER (LoadSPIByte)**

FIGURE 22: INITIATE SPI TRANSACTION (ExchangeSPI)

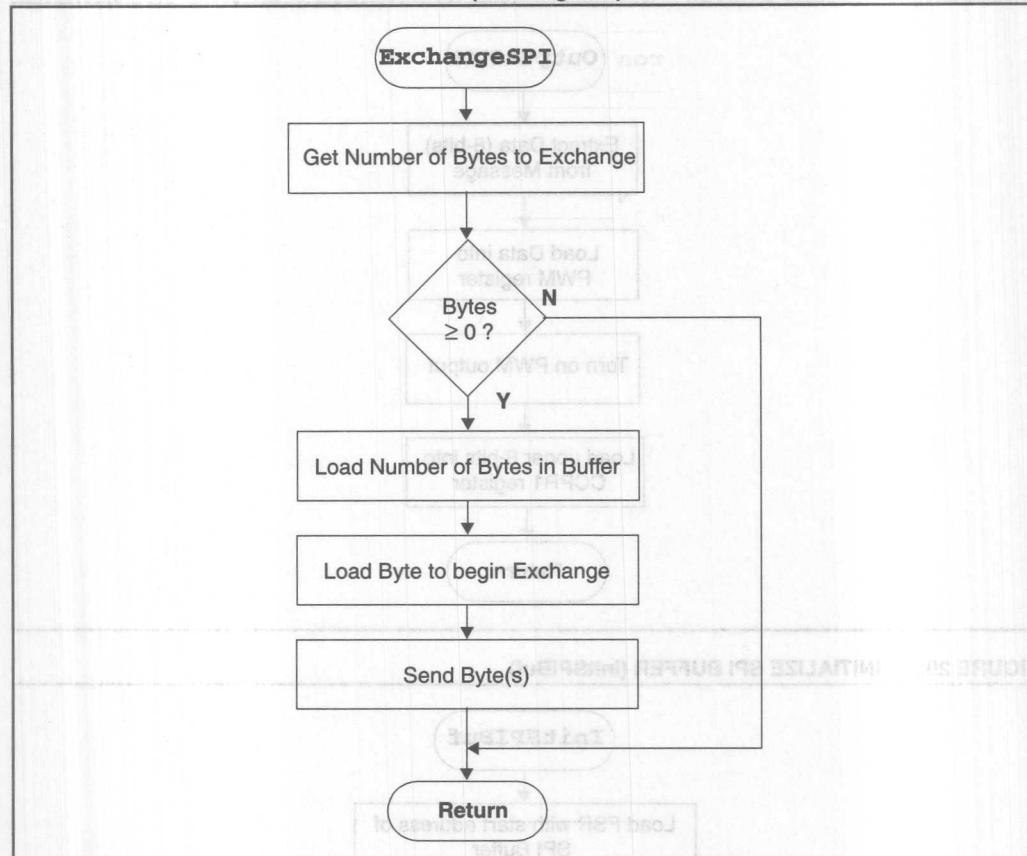


FIGURE 23: WAIT FOR SPI COMPLETION (WaitSPIExchange)

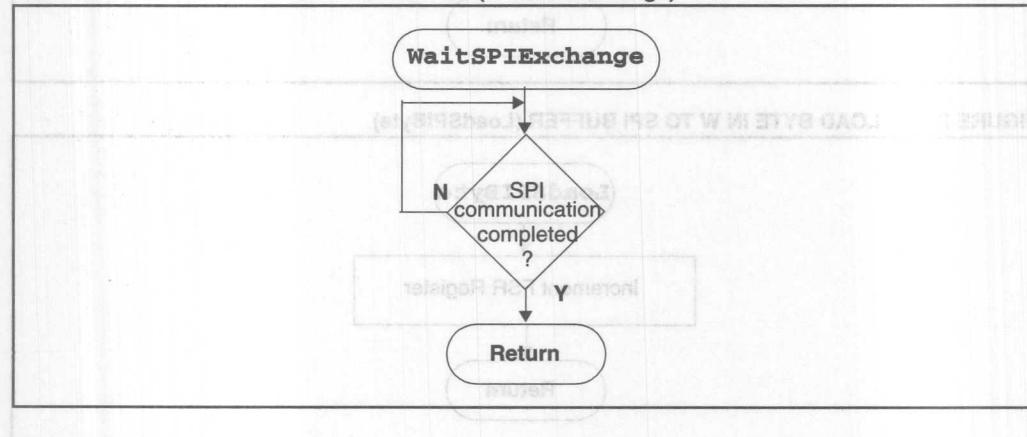
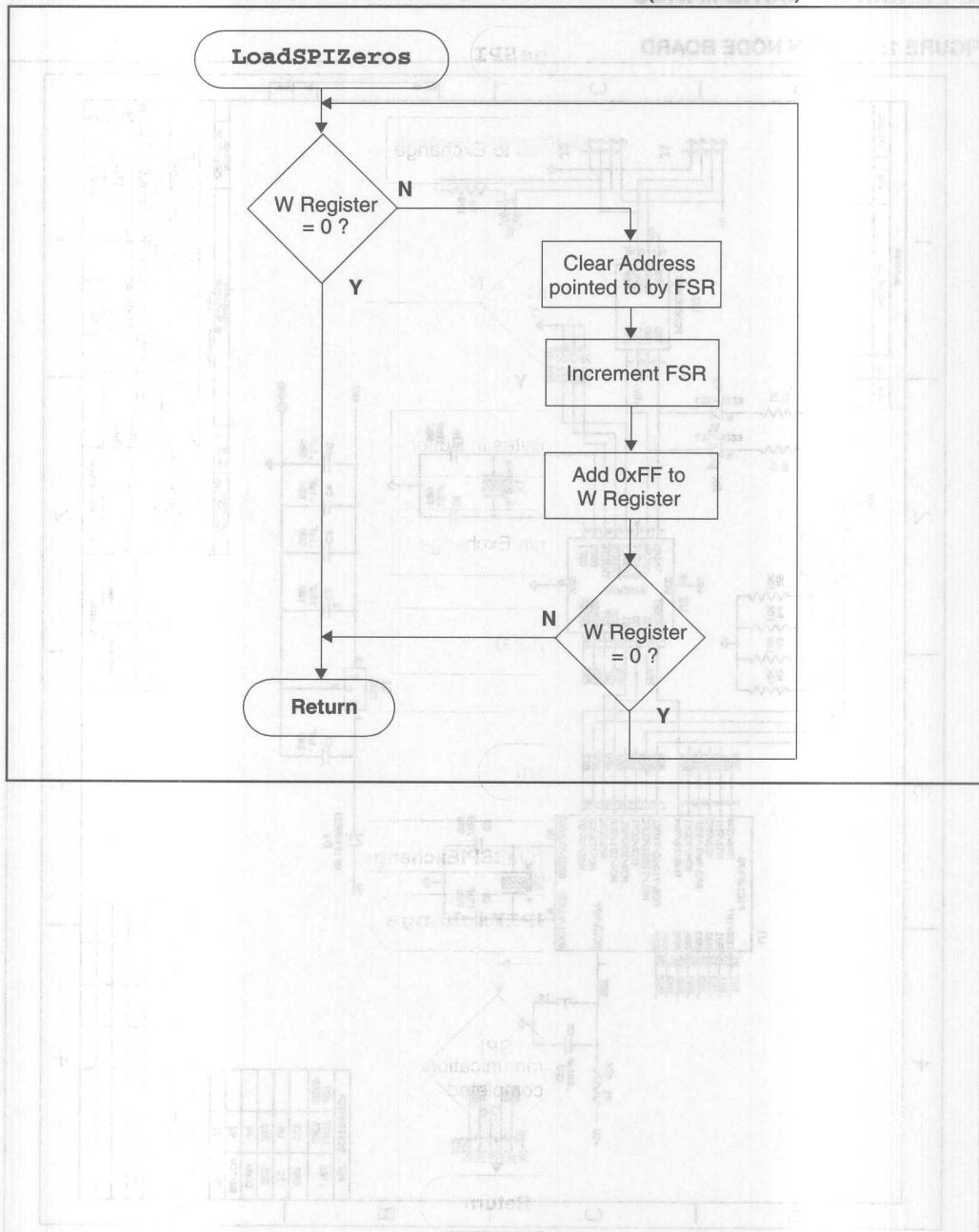


FIGURE 24: LOAD NUMBER OF ZEROS IN W TO SPI BUFFER (LoadSPIZeros)



AN212

APPENDIX A: SCHEMATICS

FIGURE 1: CAN NODE BOARD

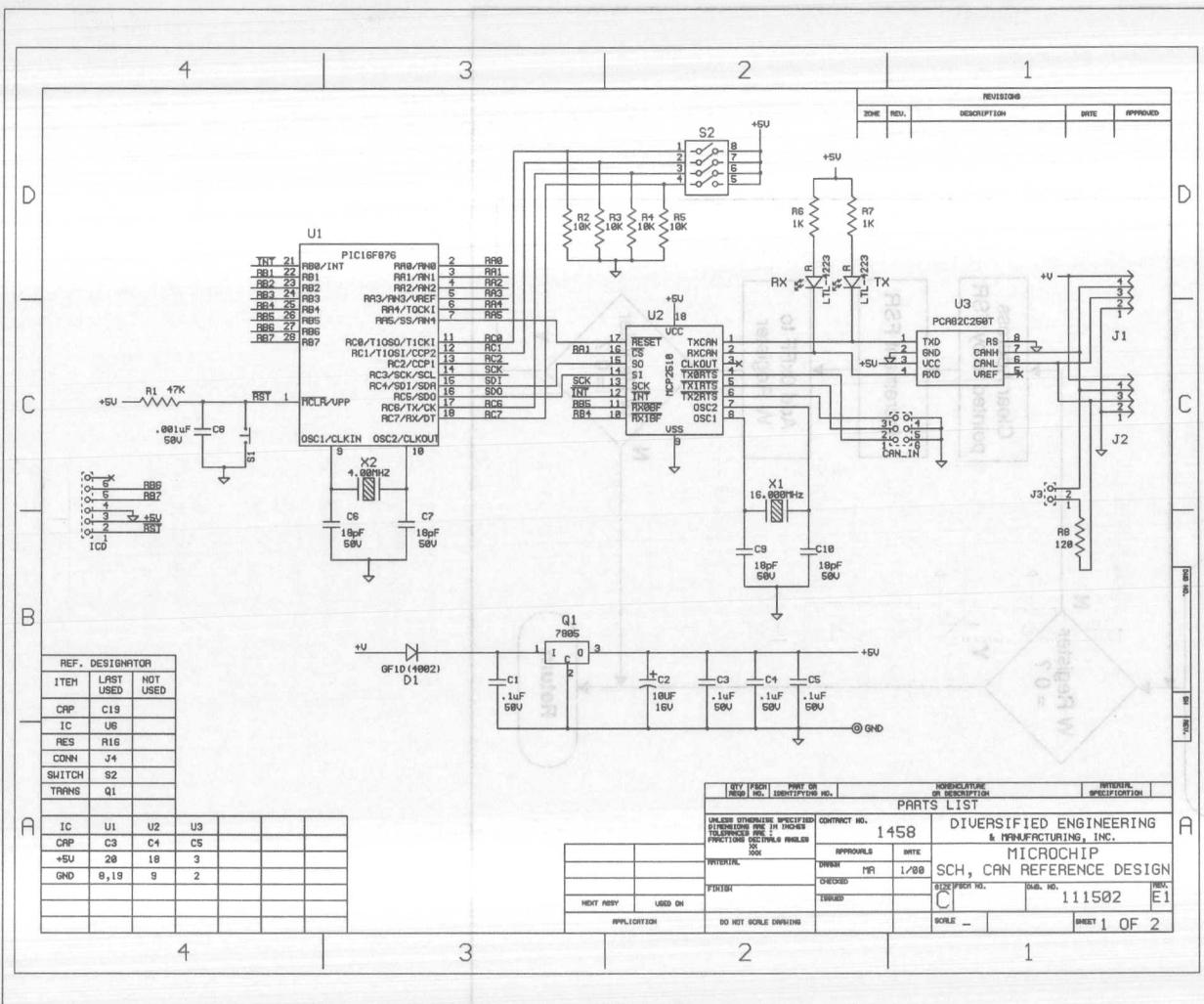
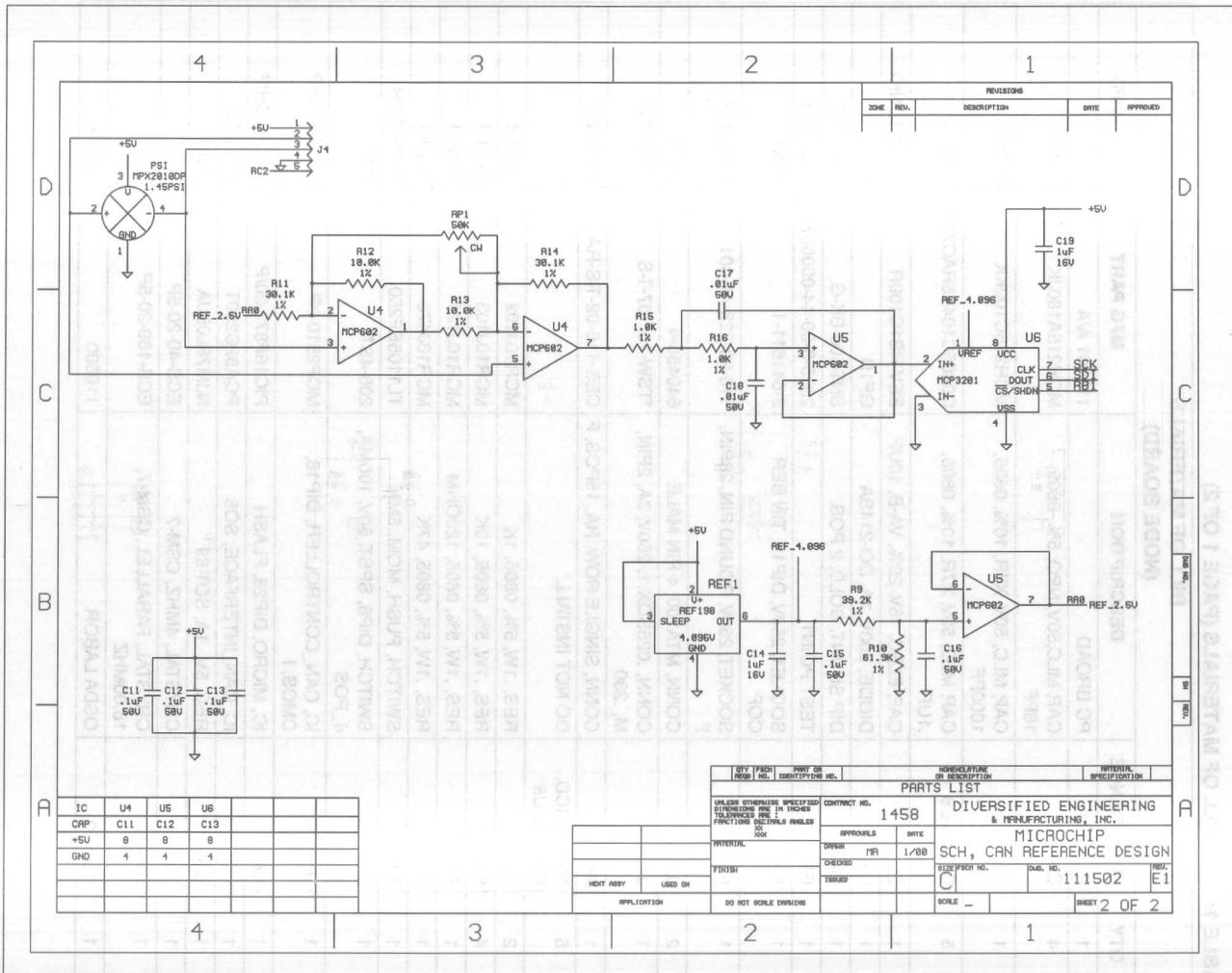


FIGURE 2: ANALOG INPUT BOARD



Notes

2 Application

AN212

APPENDIX B: BILL OF MATERIALS

TABLE 1: BILL OF MATERIALS (PAGE 1 OF 2)

BILL OF MATERIALS (NODE BOARD)				
QTY	REFERENCE	DESCRIPTION	MFG PART	MFG
1	PC BRD	PC BROAD	110501 A/A	
4	C9-C12	CAP, MLC,50V, NPO, 5%, 0805, 18PF	MCH215A180JK	ROHM
1	C8	CAP, MLC, 50V, X7R, 10%, 0805, 1000PF	MCH215C102KK	ROHM
5	C1, C3-C6	CAP, MLC, 50V, X7R, 10%, 0805, .1UF	C0805C104K5RAC7210	KEMET
1	C2	CAP, ELEC, 16V, 20%, VA-B, 10UF	EEV-HB1C100R	PANASONIC
1	D1	DIODE, 200V, 1A, DO-214BA	GF1D	GI
1	(J4)	DIP SHUNT, GOLD, 2 POS.	SNT-100-BK-G	SAMTEC
1	GND	TEST POINT	2305-3-00-44-0000070	MILLMAX
1	(U6)	SOCKET, 250V, DIP18, TIN BER COP	2-641611-1	AMP
1	(U4)	SOCKET, 250V, ROUND PIN, 28PIN, F	110-99-328-41-001	MILLMAX
2	J1, J2	CONN, MTA100, 4 PIN MALE	640456-4	AMP
1	J4	CONN, .025SQX.1, 250V, 3A, 2PIN, M, .230	"TSW-102-07-T-S	SAMTEC
1	J3	CONN, SINGLE ROW, RA, 15POS, F	CES-115-02-T-S-RA	SAMTEC
5	RX, TX, ICD, CAN-IN, U5	DO NOT INSTALL		
2	R6, R7	RES, .1W, 5%, 0805, 1K	MCR10J102	ROHM
4	R2-R5	RES, .1W, 5%, 0805, 10K	MCR10J103	ROHM
1	R8	RES, .1W, 5%, 0805, 120OHM	MCR10J121	ROHM
1	R1	RES, .1W, 5%, 0805, 47K	MCR10J473	ROHM
1	S1	SWITCH, PUSH, MOM, 6MM	TL1105EF250	E-SWITCH
1	S4	SWITCH, DIP8, SPST, 50V, 100MA, 4_POS	206-4ST	CTS
1	U6	IC, CAN_CONTROLLER, DIP18, CMOS, I	MCP2510-I/P	MICROCHIP
1	U4	IC, MICRO, DIP28, FLASH	PIC16F873-20/P	MICROCHIP
1	U7	IC, CAN_INTERFACE, SO8	PCA82C250T	PHILIPS
1	Q5	REG, 5V, .1A, SOT-89	NJM78L05UA	NJR
1	X2	CRYSTAL, 4MHZ, CSM-7	ECS-40-20-5P	ECS
1	X1	CRYSTAL, PARALLEL, CSM-7, 16.00MHZ	ECS-160-20-5P	ECS
1	110500	OSDA LABOR	110500	OSDA

TABLE 2: BILL OF MATERIALS (PAGE 2 OF 2)

BILL OF MATERIALS ANALOG INPUT BOARD				
QTY	REFERENCE	DESCRIPTION	MFG PART	MFG
1	PCB	PCB, CAN-NE ANALOG IN	110511 A/A	PCB
2	C6, C10	CAP, MLC, 16V, Y5V, +80-20, 0805, 1UF	MCH213F105ZP	ROHM
2	C5, C13	CAP, MLC, 50V, X7R, 10%, 0805, .01UF	MCH215C103KK	ROHM
7	C1-C4, C8, C11, C12	CAP, MLC, 50V, X7R, 10%, 0805, .1UF	C0805C104K5RAC721 0	KEMET
1	C7	CAP, ELEC, 16V, 20%, VA-B, 10UF	EEV-HB1C100R	PANASONIC
1	D1	DIODE, 200V, 1A, DO-214BA	GF1D	GI
1	TB1	TERM BLOCK, TH, 5POS, .1PITCH	1725685	PHEONIX
1	(J8)	SHUNT, DUAL, TIN, 2X2	MNT-102-BK-T	SAMTEC
1	J8	CONN, DUALROW, .025SQ, 3A, 8POS	TSW-104-07-T-D	SAMTEC
1	J3	CONN, SINGLE_ROW, RA, 15POS, M	TSW-115-08-T-S-RA	SAMTEC
2	XDUCER1, R11	DO NOT INSTALL		
2	R8, R9	RES, .1W, 1%, 0805, 1K	MCR10F1001	ROHM
5	R1, R3, R5, R6, R10	RES, .1W, 1%, 0805, 10.0K	MCR10F1002	ROHM
2	R4, R7	RES, .1W, 1%, 0805, 30.1K	MCR10F3012	ROHM
1	R13	RES, 1/10W, 1%, 0805, 39.2K	MCR10F3922	ROHM
1	R12	RES, 1/10W, 1%, 0805, 61.9K	MCR10F6192	ROHM
1	RP3	RES, POT, 1/2W, 10%, 20K	3299Y-203	BOURNS
1	RP1	RES, POT, 1/2W, 10%, 50K	3299Y-503	BOURNS
1	R2	THERMISTOR, tc-4.6, 10k, 1s .1", disc, TH	ERT-D2FHL103S	PANASONIC
1	U4	IC, MEMORY, DIP8, E2PROM, 512X8	25C040/P	MICROCHIP
1	U3	IC, ADC, DIP8, 12BIT, CMOS, +/- 2LSB	MCP3201-C/P	MICROCHIP
2	U1, U2	IC, CMOS, DIPS8, DUAL_OPAMP, LOW_POWER	MCP602/P	MICROCHIP
1	REF1	REG, VOLTAGE_REFERENCE, 4_096V, SO8, F	REF198FA	ANALOG_DE
1	VR1	REG, 5V, .1A, SOT-89	NJM78L05UA	NJR
1		CAN-NET ANALOG IN	110510	OSDA

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the "Company") for its PICmicro® Microcontroller is intended and supplied to you, the Company's customer, for use solely and exclusively on Microchip PICmicro Microcontroller products.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

APPENDIX C: SOURCE CODE

```
; ****
; Microchip CAN Reference Design
; node.asm
; Mike Richitelli
; Diversified Engineering
; 283 Indian River Road
; Orange, CT 06477
; (203) 799-7875 fax(203)799-7892
; WWW.DIVERSIFIEDENGINEERING.NET
;
; ****
; ****
; ***** TITLE " CAN_Ref Design "
;
; ****
dVersion equ 1
dRelease equ 5
;
; =====
; Transmits CAN message every 131 msec. Message contains two data bytes
; that represent a 12 bit value with least significant byte sent first.
; Cycles between three outputs:
; Pot: Value goes from 0 to 0xFF as Pot is turned clockwise.
; ID is selected from DIP switches #3 and #4 as follows:
; #3 #4 ID
; 0 0 transmission disabled
; 0 1 0x100
; 1 0 0x200
; 1 1 0x300
;
; Push button switch: Switch open => 0, Switch closed => 0xFFFF
; ID is Pot ID + 0x010
;
; Cds: Value goes from 0 to 0xFF as Pot is turned clockwise.
; ID is Pot ID + 0x020
;
; CAN messages received are assumed to be 12 bit data sent as two bytes,
; least significant byte first.
;
; The base ID for receiving CAN messages is specified by DIP switches #1 and #2:
; #1 #2 ID
; 0 0 0x000
; 0 1 0x100
; 1 0 0x200
```

```

;           1 1   0x300

; Lamp: If the message ID matches the ID selected by the DIP
; output switches the 12 bit data is used to generate a PWM
; generates where a 0 value gives a zero duty cycle and 0xFFFF gives
; a 100% duty cycle. The lamp output is proportional to
; the duty cycle. ID is Base ID
;

; LED: On if value received is >= 0x800 and off if < 0x800.
;       ID is Base ID + 0x010
;

=====

----- PIC16F876 Micro ----;

LIST P=16F876
LIST r=dec,x=on,t=off
#include "P16F876.INC"

__CONFIG
_BODEN_ON_&_CP_OFF_&_WRT_ENABLE_ON_&_PWRTE_ON_&_WDT_OFF_&_HS_OSC_&_DEBUG_OFF_&_CPD_OFF_&_LVP_OFF
_IDLOCS (dVersion<<8)|dRelease ; version: vvrr , vv - version, rr - release

-----

#include "MACROS16.INC"
#include "MCP2510.INC"
; errorlevel 0,-306,-302,-305

;***** constants

;Crystal freq 4.00 MHz, Fosc/4 = 1 uS
; Timer 1: Uses no prescale => Tic is 1 uSec
;           8 bit rollover 256 uSec
;           16 bit rollover 65.536 msec

; 8 bit timers
; TMR1L: 1 uSec tics with maximum of 1/2 rollover = 128 uSec maximum
; TMR1H: 256 uSec tics with maximum of 1/2 rollover = 32.768 msec
;maximum

; A/D selection ( value of ADCON0 )
#define dA2DRA0    B'01000000'      ; fosc/8 clk, RA0, A/D off
#define dA2DRA3    B'01011000'      ; fosc/8 clk, RA3, A/D off

; special function defines
#define _SSPEN     SSPCON,SSPEN    ; SPI enable pin muxed to SCK
;

=====

```

```

;; General control flags definitions
#define tbWork          bGenFlags1,0 ; working bit
#define tbReset         bGenFlags1,1 ; must reset
#define tbNewSPI        bGenFlags1,2 ; new SPI data available
#define tbRxMsgPend    bGenFlags1,3 ; new CAN message received
#define tbTxMsg         bGenFlags1,4 ; xmit next CAN message
#define tbRC2NowHigh   bGenFlags1,5 ; Robot PWM signal high

;***** PIN DEFINITIONS *****
#define tp2510_CS_      PORTA,1      ; CS_ for MCP2510 chip
#define tpSwitch_        PORTB,1      ; Push button switch, Open => high
#define tpLED            PORTB,2      ; LED

; Analog In
#define tpA2D_CS_        PORTB,1      ; CS_ for 3201 A2D chip
#define tpEE_CS_          PORTB,2      ; CS_ for 25040 E2

;***** LOCAL REGISTER STORAGE *****
;===== BANK 0 =====
cblock 0x20
    ; interrupt variables
    bIntSaveSt       ; save Status
    bIntSaveFSR      ; save FSR
    bIntSavPCLATH   ; interrupt storage for PCLATH
    bIntWork         ; working
    iIntWork:2        ; working

    ; general work space
    bGenFlags1        ; general control flags 1
    bGenFlags2        ; general control flags 2
    bWork             ; work byte
    bWork1            ; work byte
    iWork:2           ; work integer
    bCnt              ; work byte counter

    ; Arithmetic
    iA:2              ; 2 byte integer - receiver CAN ID minimum data word
    iB:2              ; 2 byte integer receiver CAN ID maximum data word

    ; Timer1
    iTimer1:2         ; counts Timer1 rollover
    bGenClk           ; general clock
    bXmitClk          ; Countdown to xmit next message

    ; In/Out variables
    iA2DValue:2       ; 12 bit or 8 bit A2D value
    bPWMValue         ; 8 bit PWM value
    iRecValue:2        ; 12 bit received value

    ; general control
    bSwXmitID        ; ID for transmission from DIP switch
    bBaseRecID        ; ID for reception from DIP switch
    bRecIDNext        ; Rec Base ID + 1
    bXmitID           ; ID for transmission of next msg
    bNextMsgType      ; Select next msg

    ; Received CAN message

```

```

    iRecID_L          ; ID of received message (3 bits left
;justified)
    iRecID_H          ; ID of received message (8 bits left w/o) R16H
;justified)
    bRecCount         ; number of bytes received
    pRecDataBase:8   ; received data
    ; Low level SPI interface
    b2510RegAdr     ; Register address
    b2510RegData     ; Data sent/received
    b2510RegMask     ; Bit Mask

    ; following used in interrupt
    bSPICnt          ; # bytes remaining to receive - (and w/o) R16H
    pSPIBuf           ; Pointer into buffer
    pSPIBufBase:12   ; Base of SPI receive/xmit buffer

    endc

; storage for interrupt service routine
; W saved in one of these locations depending on the page selected
; at the time the interrupt occurred

bIntSaveW0 equ 0x7F      ; interrupt storage for W

;===== BANK 1 =====
bIntSaveW1 equ 0xFF      ; interrupt storage for W

;***** LOCAL MACROS *****
;***** LOCAL MACROS *****
;***** LOCAL MACROS *****

;
; Shift left 2 byte integer once.
iShiftL macro iVar
    bcf _C           ; clear carry bit
    rlf iVar,F
    rlf iVar+1,F
    endm

; Shift right 2 byte integer once.
iShiftR macro iVar
    bcf _C           ; clear carry bit
    rrf iVar+1,F
    rrf iVar,F
    endm

; Increment 2 byte integer
intInc macro iVar
    incf iVar,F
    skipNZ
    incf iVar+1,F
    endm

;

; Set TRM1H 8 bit clock
; TMR1H: 256 uSec ticks with maximum of 1/2 rollover = 32.768 msec ;
; maximum
;

Set1HClock macro bClk,Value
    movfw TMR1H
    addlw Value
    movwf bClk
    endm

```

AN212

```

;***** Begin Program Code *****
;***** HardStart *****
        ORG      0x0          ;memory @ 0x0
        nop          ;nop ICD!!
        goto     HardStart

        ORG      04h          ;Interrupt Vector @ 0x4
;***** Interrupt service routine - must be at location 4 if page 1 is used *****
; Context save & restore takes ~20 instr
;***** Global int bit, GIE, has been reset. W in 1,0 and dly bdsd qd need
; W saved in bIntSaveW0 or bIntSaveW1 depending on the bank
; selected at
; the time the interrupt occurred.
        movwf   bIntSaveW0      ; save W in either of two locations
                                ; depending on bank currently
;selected
; only way to preserve Status bits (since movf sets Z) is with a swapf
; swapf command now
        swapf   STATUS,W       ; Status to W with nibbles swapped
        BANK0
        movwf   bIntSaveSt      ; save Status to bIntSaveSt
        movfw   FSR
        movwf   bIntSaveFSR     ; save FSR with original stored
        movf    PCLATH,W        ; save PCLATH with original stored
        movwf   bIntSavPCLATH   ; interrupt storage for PCLATH
        clrf    PCLATH          ; set to page 0
; Must determine source of interrupt
; SPI interrupt
        btfsc   _SSPIF         ; SPI interrupt
        goto    IntSPI
        jmpSet  _TMR1IF,jIntTimer1 ; Timer1 overflow interrupt flag
; unknown
; restore registers and return
        IntReturn
        BANK0
        movf    bIntSavPCLATH,W ; interrupt storage for PCLATH
        movwf   PCLATH
        movf    bIntSaveFSR,W    ; restore FSR
        movwf   FSR
        swapf  bIntSaveSt,W     ; get swapped Status (now unswapped)
        movwf   STATUS           ; W to Status ( bank select restored )
        swapf  bIntSaveW0,F      ; swap original W in place
        swapf  bIntSaveW0,W      ; now load and unswap ( no status
;change)
        retfie                 ; return from interrupt

```

```

; Look up ID associated with bits 0,1 in W
RxIDTable addwf PCL,F ;Jump to char pointed to in W reg
; ( adds 5bits from PCLATH )
    retlw 0x00 ; 0
    retlw 0x20 ; 1
    retlw 0x10 ; 2
    retlw 0x30 ; 3

RxIDTable_End
#endif

;if ( (RxIDTable & 0xF00) != (RxIDTable_End & 0xF00) )
    MESSG "Warning - Table crosses page boundary in computed jump"
#endif

; Look up ID associated with bits 0,1 in W
TxIDTable addwf PCL,F ;Jump to char pointed to in W reg
; ( adds 5bits from PCLATH )
    retlw 0xFF ; 0
    retlw 0x20 ; 1
    retlw 0x10 ; 2
    retlw 0x30 ; 3

TxIDTable_End
#endif

;if ( (TxIDTable & 0xF00) != (TxIDTable_End & 0xF00) )
    MESSG "Warning - Table crosses page boundary in computed jump"
#endif

;***** LIBRARY STORAGE & FUNCTIONS *****
#include "CanLib.asm" ; basic MCP2510 interface routines
#include "a2d3201.asm" ; MCP3201 AD routines

;***** Local Interrupt Handlers *****

;*****
jIntTimer1
; Timer1 rollover interrupt.
;
;*****
jIntTimer1 ; Timer1 overflow interrupt flag
    bcf _TMR1IF ; timer1 rollover interrupt flag
    intInc iTimer1

    jmpFeqZ bXmitClk,IntReturn

    decfsz bXmitClk,F ; Countdown to xmit next message
    goto IntReturn

; Countdown to xmit next message
    bsf tbTxMsg ; xmit next CAN msg
    goto IntReturn

```

```

;*****
;IntSPI
;
; A single buffer, at pSPIBufBase, is used for both SPI receive and
; transmit. When a byte is removed from the buffer to transmit it is
; replaced by the byte received.
;
; When here the buffer pointer, pSPIBuf, points to the last byte loaded
; for transmission. This is the location that the received byte will be stored.
;
; When here the count, bSPICnt, contains the number of bytes remaining
; to be received. This is one less then the number remaining to be
; transmitted. When bSPICnt reaches zero the transaction is complete.
;
;*****
        bcf      _SSPIF           ; clear interrupt flag
        incf      pSPIBuf,F
        movfw   SSPBUF            ; get data & clear buffer flag
        movwf   INDF             ; put it into SPI buffer
        decfsz  bSPICnt,F
        goto    jIntSPII          ; More bytes to send
;
        bsf      tp2510_CS_       ; CS_ for MCP2510 chip
        goto    IntReturn
;
jIntSPII
        incf      FSR,F
        movfw   INDF             ; get byte from buffer
        movwf   SSPBUF            ; send it
        goto    IntReturn
;
```

AN212

```

;***** HardStart *****

HardStart
    call      Init

    ; Make sure no chips are selected
    bsf      tp2510_CS_11; CS_ for MCP2510 chip

    ; Read DIP switch and create bBaseRecID and bSwXmitID
    movfw   PORTC
    andlw  0x03
    call    RxIDTable
    movwf  bBaseRecID
    movwf  bSwXmitID

    ; Xmit ID bits are at pins 6,7 of PORTC and are logic low = 1
    swapf  PORTC,W
    movwf  bSwXmitID
    rrfbSwXmitID,F
    rrfbSwXmitID,W
    andlw  0x03
    call    TxIDTable
    movwf  bSwXmitID

    ; ----- One time calculations -----
    call    InitSPIPort

    ; Wait 28 mS for MCP2510 to initialize ( there is no significance to 28 ms -
    ; we just selected a large time since time is not critical)
    Set1HClock bGenClk,100 ; 277.77 uSec ticks

jInit5
    jmp1HNotYet bGenClk,jInit5

    ; Setup all MCP2510 registers
    call      Init2510

    bsf      tbTxMsa          ; xmit flag

```

```

;; -----
;; ----- MAIN LOOP -----
;;

jMainLoop clrwdt

;===== XMIT CODE =====

;jmpClr tbTxMsg,jMain10 ; not time to xmit next CAN msg yet
;yet bcf tbTxMsg ; reset xmit flag
; Reload counter
; movlw 2 ; 65 mS units
; movwf bXmitClk **** ; Countdown to xmit next message
;jmpFegl bXmitID,0xFF,jMain10 ; Transmission turned off to MSG ready
; Time to xmit next CAN message. Select source of message and ID
; to use for transmission
; <<<< Analog Input Board >>>
;jmpFegl bNextMsgType,0,Xmit3201
;jmpFegl bNextMsgType,1,XmitRA0
;goto jMain8
;***** POT *****
Xmit3201
;call Read3201 ; read 3201 AD
;movfw bSwXmitID ; Use DIP Tx address for transmission
;movwf bXmitID
;incf bNextMsgType,F ; Next time use next source and ID
;goto jMain8
XmitRA0
;movlw dA2DRA0 ; fosc/8 clk, RA0, A/D off
;call ReadA2D ; Read A/D port in W
;movfw bSwXmitID ; Use DIP Tx address for transmission
;addlw 0x01
;movwf bXmitID
;clr f bNextMsgType ; clear next message
;goto jMain8
jMain8
; Wait for pending messages to be sent (ALL BUFFERS)
;bl2bV 0x08,b2510RegMask
;movlw TXB0CTRL
;call WaitANDeqZ
; Send CAN message with
; ID = bXmitID
; Two data bytes: iA2DValue,iA2DValue+1
;SPI_WriteV TXB0SIDH,bXmitID ; Message ID
;SPI_WriteL TXB0SIDL,0x00 ; Send message - lower bits 0
;SPI_WriteL TXB0DLC,0x02 ; 2 data bytes
; Send least significant byte first
;SPI_WriteV TXB0D0,iA2DValue
;SPI_WriteV TXB0D1,iA2DValue+1
;SPI_Rts RTS0 ; Transmit buffer 0

```



```

;*****
;ReadA2D
; This functions reads analog input and stores the result in ia2DValue as a 12 bit value. Value in W is used to set the ADCON0 to select correct port.
;*****
ReadA2D

    ; setup A/D to select port, etc
    movwf    ADCON0

    ; turn on A/D
    bsf     ADCON0,0      ;A/D on
    ; allow 50us for settling
    movlw   25
    movwf   bCnt
    ReadAD10
        decfsz bCnt,F
        goto    ReadAD10
    ; begin conversion
    bsf     ADCON0,2      ; GO
ReadAD20 jmpSet  ADCON0,2,ReadAD20  ; wait for done bit

    movf    ADRES,W
    movwf   ia2DValue
    clrf    ia2DValue+1

    ; Convert to 12 bit
    iShiftL ia2DValue
    iShiftL ia2DValue
    iShiftL ia2DValue
    iShiftL ia2DValue
    return

```

```
;*****  
;ParseCAN <<<INPUTS>>>  
;  
; Parse message. Assumes message is two byte 12 bit data.  
; Uses bBaseRecID and bRecIDNext to accept message for PWM if it is equal to  
; output.  
;*****  
ParseCAN  
  
;***** Analog In Board *****  
  
        movlw    b'00001111'  
        andwf   iRecID_H,W  
        movwf   iA  
        jmpFegl ia,0x0,IDX0      ;check for ID x0  
        goto    jParCANRet  
  
IDX0  
;to lamp  
;; 12 bits of data  
        bV2bV   pRecDataBase,iRecValue  
        bV2bV   pRecDataBase+1,iRecValue+1  
  
;; new PWM value pending  
        bV2bV   iRecValue,iA  
        bV2bV   iRecValue+1,iA+1      ; 12 bit received value  
  
;; convert to 8 for PWM out  
        iShiftR iA  
        iShiftR iA  
        iShiftR iA  
        iShiftR iA  
  
;; Convert to 8 bit  
        bV2bV   iA,bPWMValue  
        call    OutputPWM  
        goto    jParCANRet  
  
jParCANRet  
        return
```

```

;*****  

;Init2510  

;* Function: Init_MCP2510()  

;* Place MCP2510 initialization here...  

;*****  

Init2510  

    ;; Reset MCP2510  

        call      Reset2510  

    ;; set CLKOUT prescaler to div by 4  

        bL2bV    0x03,b2510RegMask  

        bL2bV    0x02,b2510RegData  

        movlw    CANCTRL  

        call      BitMod2510  

;Set physical layer configuration  

;  

;    Fosc = 16MHz  

;    BRP      = 7 (divide by 8)  

;    Sync Seg   = 1TQ  

;    Prop Seg   = 1TQ  

;    Phase Seg1 = 3TQ  

;    Phase Seg2 = 3TQ  

;  

;    TQ = 2 * (1/Fosc) * (BRP+1)  

;    Bus speed = 1/(Total # of TQ) * TQ  

;  

        SPI_WriteL CNF1,0x07           ; set BRP to div by 8  

;#define BTLMODE_CNF3    0x80  

;#define SMPL_1X       0x00  

;#define PHSEG1_3TQ    0x10  

;#define PRSEG_1TQ     0x00      (1 TQ) divide into 4 TQ :  

        SPI_WriteL CNF2,0x90  

;#define PHSEG2_3TQ    0x02  

        SPI_WriteL CNF3,0x02  

;  

    ; Configure Receive buffer 0 Mask and Filters  

    ; Receive buffer 0 will not be used  

        SPI_WriteL RXM0SIDH,0xFF  

        SPI_WriteL RXM0SIDL,0xFF  

        SPI_WriteL RXF0SIDH,0xFF  

        SPI_WriteL RXF0SIDL,0xFF  

        SPI_WriteL RXF1SIDH,0xFF  

        SPI_WriteL RXF1SIDL,0xFF  

    ; Configure Receive Buffer 1 Mask and Filters  

        SPI_WriteL RXM1SIDH,0xFF  

        SPI_WriteL RXM1SIDL,0xE0  

    ; Initialize Filter 2 to match x0 bBaseRecID from DIP switch  

        SPI_WriteV RXF2SIDH,bBaseRecID ; = revolvo x0 to maximum data size  

        SPI_WriteL RXF2SIDL,0x00      ; Make sure EXIDE bit (bit 3)  

;is set correctly in filter  

    ; Initialize Filter 3 to match x1 from DIP switch  

        incf    bBaseRecID,F  

        SPI_WriteV RXF3SIDH,bBaseRecID  

        SPI_WriteL RXF3SIDL,0x00  

    ; Initialize Filter 4 to match x2 from DIP switch

```

AN212

```
incf      bBaseRecID,F
SPI_WriteV RXF4SIDH,bBaseRecID
SPI_WriteL RXF4SIDL,0x00

; Initialize Filter 5 to match x3 from DIP switch
incf      bBaseRecID,F
SPI_WriteV RXF5SIDH,bBaseRecID
SPI_WriteL RXF5SIDL,0x00

movlw    b'11110000'
andwf    bBaseRecID,F

; Disable all MCP2510 Interrupts
bL2bV    0x00,b2510RegData
movlw    CANINTE
call     Wrt2510Reg

; Sets normal mode
call     SetNormalMode
return

;*****ProcessSPI*****
;ProcessSPI
skipSet  bSPICnt,2
; buffer not full yet
return

; disable SPI interrupt
BANK1
bcf     _SSPIE_P ; SSP int enable (BANK 1)
BANK0

; enable SPI
BANK1
bsf     _SSPIE_P ; SSP int enable (BANK 1)
BANK0
return

;*****WaitMSec*****
;WaitMSec
; Delay W number of Msec Routines (255 max)
; Actually slightly larger than 1 mS
;*****WaitMSec0*****
WaitMSec
movwfbCnt ;store Msec -> bCnt

jWaitMSec0
clrwdt ;clear wdt

; TMR1H: 256 uSec ticks with maximum of 1/2 rollover = 32.768 msec
;maximum
Set1HClock bGenClk,4 ; 256 uS

jWaitMSec1
jmp1HNotYet bGenClk,jWaitMSec1

decfsz  bCnt,F
goto    jWaitMSec0
return
```

```

;
;***** Initialize *****
;Init
;    Initialize
;
;***** Initialize *****
Init
    clrwdt           ; required before changing wdt to timer0
;
;; clear peripheral interrupts
BANK1
    clrf    PIE1_P
;
;; OPTION_REG: PortB Pullups on.
;; no prescale for WDT -> should always > 7 mSec ( 18 mS nominal)
;; Timer 0: Use 64 prescale for 0.27127 * 64 = 17.361 uSec tics
        movlw   B'01000101'      ; Timer0 prescale 64
        movwf   OPTION_REG_P
;
;; clear bank 0
    movlw   0x20
    movwf   FSR
jInitClr1 clrf   INDF
    incf   FSR,F
    jmpClr  FSR,7,jInitClr1
;
;; clear bank 1
    movlw   0xA0
    movwf   FSR
jInitClr2 clrf   INDF
    incf   FSR,F
    jmpSet  FSR,7,jInitClr2
;
    call    InitIO          ; initialize IO of microcontroller
;
;; configure Timer1:
    BANK0
        movlw   B'00000001'      ; Prescale = 1, Timer enabled
        movwf   T1CON
;
BANK1
    bsf     _TMR1IE_P       ; timer1 rollover/interrupt enable
;(page 1)
    BANK0
;
;; init output PWM1 ( uses timer2 )
    BANK0
        movlw   B'00000100'      ; prescale of 1, internal clk, enable
;
,timer2
    movwf   T2CON
        ; load PWM counter(PR2) with 0x3F ( 8 bit high res mode)
        ; this gives a 15.625KHz signal with 4MHz crystal
    BANK1
    movlw   0x3F
    movwf   PR2_P
    BANK0
;
;; for testing
    clrf    TMR1L
    clrf    TMR1H

```

AN212

```

; ; turn on interrupts
BANK0
    movlw B'11000000'; Enable interrupts ( Periphrales
;only )
    movwf INTCON

    return

; ;----- Output of the pinpads enabled Seripot :
; ; INITIALIZE I/O OF MICROCONTROLLER

InitIO
    BANK1
        movlw b'00000100'; turn on A/D conversion RA0, RA1, RA3
        movwf ADCON1_P

; ; Port A
; ;      0 in    <*>A2D input POT<*>
; ;      1 out(1) MCP2510 chip select also analog GND
; ;      2 in     <*>open<*>
; ;      3 in     <*>open<*>
; ;      4 in     <*>open<*>
; ;      5 out(1) RST MCP2510

        BANK0
        movlwB'00000010'; initialize Port A outputs
        movfw PORTA
    BANK1
        movlw B'11111101'
        movwf TRISA_P      ; set Port A

; ; Port B
; ;      0 in     Interrupt from MCP2510
; ;      1 out(1) <*>CS' MCP3201<*>
; ;      2 out(1) <*>CS' 25C04<*> also to CS expansion
; ;      3 in     <*>open<*>
; ;      4 in     <*>open<*>
; ;      5 in     RX0BF from MCP2510
; ;      6 in     ICD
; ;      7 in     ICD

        BANK0
        movlwB'00000110'; initialize Port B outputs
        movfw PORTB
    BANK1
        movlw B'111111001'
        movwf TRISB_P      ; set Port B

; ; Port C
; ;      0 in     DIP #1
; ;      1 in     DIP #2
; ;      2 out(0) <*>PWM Out<*>
; ;      3 out(0) SPI clock - master
; ;      4 in     SPI data in
; ;      5 out(0) SPI data out
; ;      6 in     DIP #3
; ;      7 in     DIP #4

        BANK0
        movlw B'00000000'
        movwf PORTC
    BANK1
        movlw B'11010011'

```

```

movwf    TRISC_P      ; set Port C

BANK0
return

ifdef ROBUST
; robust design - force WDT reset
    FILL (goto WDTReset1),(0xFFFF-$)
WDTReset1 goto    WDTReset1
endif

END

```

00x0	0000 00000000
10x0	0000 00000000
20x0	0000 00000000
30x0	0000 00000000
40x0	0000 00000000
50x0	0000 00000000
60x0	0000 00000000
70x0	0000 00000000
80x0	0000 00000000
90x0	0000 00000000
A0x0	0000 00000000
B0x0	0000 00000000
C0x0	0000 00000000
D0x0	0000 00000000
E0x0	0000 00000000
F0x0	0000 00000000
01x0	0000 00000000
11x0	0000 00000000
21x0	0000 00000000
31x0	0000 00000000
41x0	0000 00000000
51x0	0000 00000000
61x0	0000 00000000
71x0	0000 00000000
81x0	0000 00000000
91x0	0000 00000000
A1x0	0000 00000000
B1x0	0000 00000000
C1x0	0000 00000000
D1x0	0000 00000000
E1x0	0000 00000000
F1x0	0000 00000000
02x0	0000 00000000
12x0	0000 00000000
22x0	0000 00000000
32x0	0000 00000000
42x0	0000 00000000
52x0	0000 00000000
62x0	0000 00000000
72x0	0000 00000000
82x0	0000 00000000
92x0	0000 00000000
A2x0	0000 00000000
B2x0	0000 00000000
C2x0	0000 00000000
D2x0	0000 00000000
E2x0	0000 00000000
F2x0	0000 00000000
03x0	0000 00000000
13x0	0000 00000000
23x0	0000 00000000
33x0	0000 00000000
43x0	0000 00000000
53x0	0000 00000000
63x0	0000 00000000
73x0	0000 00000000
83x0	0000 00000000
93x0	0000 00000000
A3x0	0000 00000000
B3x0	0000 00000000
C3x0	0000 00000000
D3x0	0000 00000000
E3x0	0000 00000000
F3x0	0000 00000000

AN212

```
;-- MCP2510.INC
; Description: This file contains the definitions for the MicroChip
; standalone CANbus controller.
;
; 07/17/99 JFF Original Version
; 09/11/99 JCT Modified for ASM
;-----  
  
#define RXF0SIDH      0x00
#define RXF0SIDL      0x01
#define RXF0EID8      0x02
#define RXF0EID0      0x03
#define RXF1SIDH      0x04
#define RXF1SIDL      0x05
#define RXF1EID8      0x06
#define RXF1EID0      0x07
#define RXF2SIDH      0x08
#define RXF2SIDL      0x09
#define RXF2EID8      0x0A
#define RXF2EID0      0x0B
#define BFPCTRL       0x0C
#define TXRTSCTRL    0x0D
#define CANSTAT       0x0E
#define CANCTRL       0x0F  
  
#define RXF3SIDH      0x10
#define RXF3SIDL      0x11
#define RXF3EID8      0x12
#define RXF3EID0      0x13
#define RXF4SIDH      0x14
#define RXF4SIDL      0x15
#define RXF4EID8      0x16
#define RXF4EID0      0x17
#define RXF5SIDH      0x18
#define RXF5SIDL      0x19
#define RXF5EID8      0x1A
#define RXF5EID0      0x1B
#define TEC           0x1C
#define REC           0x1D
#define CANSTAT1     0x1E
#define CANCTRL1     0x1F  
  
#define RXM0SIDH      0x20
#define RXM0SIDL      0x21
#define RXM0EID8      0x22
#define RXM0EID0      0x23
#define RXM1SIDH      0x24
#define RXM1SIDL      0x25
#define RXM1EID8      0x26
#define RXM1EID0      0x27
#define CNF3          0x28
#define CNF2          0x29
#define CNF1          0x2A
#define CANINTE       0x2B
#define CANINTF       0x2C
#define EFLG          0x2D
#define CANSTAT2     0x2E
#define CANCTRL2     0x2F  
  
#define TXB0CTRL      0x30
#define TXB0SIDH      0x31
#define TXB0SIDL      0x32
#define TXB0EID8      0x33
#define TXB0EID0      0x34
```

```

#define TXB0DLC      0x35
#define TXB0D0      0x36
#define TXB0D1      0x37
#define TXB0D2      0x38
#define TXB0D3      0x39
#define TXB0D4      0x3A
#define TXB0D5      0x3B
#define TXB0D6      0x3C
#define TXB0D7      0x3D
#define CANSTAT3    0x3E
#define CANCTRL3    0x3F

#define TXB1CTRL     0x40
#define TXB1SIDH    0x41
#define TXB1SIDL    0x42
#define TXB1EID8    0x43
#define TXB1EIDO    0x44
#define TXB1DLC     0x45
#define TXB1D0      0x46
#define TXB1D1      0x47
#define TXB1D2      0x48
#define TXB1D3      0x49
#define TXB1D4      0x4A
#define TXB1D5      0x4B
#define TXB1D6      0x4C
#define TXB1D7      0x4D
#define CANSTAT4    0x4E
#define CANCTRL4    0x4F

#define TXB2CTRL     0x50
#define TXB2SIDH    0x51
#define TXB2SIDL    0x52
#define TXB2EID8    0x53
#define TXB2EIDO    0x54
#define TXB2DLC     0x55
#define TXB2D0      0x56
#define TXB2D1      0x57
#define TXB2D2      0x58
#define TXB2D3      0x59
#define TXB2D4      0x5A
#define TXB2D5      0x5B
#define TXB2D6      0x5C
#define TXB2D7      0x5D
#define CANSTAT5    0x5E
#define CANCTRL5    0x5F

#define RXB0CTRL     0x60
#define RXB0SIDH    0x61
#define RXB0SIDL    0x62
#define RXB0EID8    0x63
#define RXB0EIDO    0x64
#define RXB0DLC     0x65
#define RXB0D0      0x66
#define RXB0D1      0x67
#define RXB0D2      0x68
#define RXB0D3      0x69
#define RXB0D4      0x6A
#define RXB0D5      0x6B
#define RXB0D6      0x6C
#define RXB0D7      0x6D
#define CANSTAT6    0x6E
#define CANCTRL6    0x6F

#define RXB1CTRL     0x70
#define RXB1SIDH    0x71

```

#define RXB1SIDL	0x72	0x00	TXB1 SIDL
#define RXB1ID8	0x73	0x01	TXB1 ID8
#define RXB1EID0	0x74	0x02	TXB1 EID0
#define RXB1DLC	0x75	0x03	TXB1 DLC
#define RXB1D0	0x76	0x04	TXB1 D0
#define RXB1D1	0x77	0x05	TXB1 D1
#define RXB1D2	0x78	0x06	TXB1 D2
#define RXB1D3	0x79	0x07	TXB1 D3
#define RXB1D4	0x7A	0x08	TXB1 D4
#define RXB1D5	0x7B	0x09	TXB1 D5
#define RXB1D6	0x7C	0x0A	TXB1 D6
#define RXB1D7	0x7D	0x0B	TXB1 D7
#define CANSTAT7	0x7E	0x0C	CANSTAT7
#define CANCTRL7	0x7F	0x0D	CANCTRL7
 ;; Bit definitions		0x0E	CANCTRL7
 ;; Bit definitions		0x0F	BFPCTRL
#define trB1BFS	BFPCTRL, 5	0x10	BFPCTRL, 5
#define trB0BFS	BFPCTRL, 4	0x11	BFPCTRL, 4
#define trB1BFE	BFPCTRL, 3	0x12	BFPCTRL, 3
#define trB0BFE	BFPCTRL, 2	0x13	BFPCTRL, 2
#define trB1BFM	BFPCTRL, 1	0x14	BFPCTRL, 1
#define trB0BFM	BFPCTRL, 0	0x15	BFPCTRL, 0
 ;; Bit definitions		0x16	TXRTSCTRL
#define trB2RTS	BFPCTRL, 5	0x17	BFPCTRL, 5
#define trB1RTS	BFPCTRL, 4	0x18	BFPCTRL, 4
#define trB0RTS	BFPCTRL, 3	0x19	BFPCTRL, 3
#define trB2RTSM	BFPCTRL, 2	0x1A	BFPCTRL, 2
#define trB1RTSM	BFPCTRL, 1	0x1B	BFPCTRL, 1
#define trB0RTSM	BFPCTRL, 0	0x1C	BFPCTRL, 0
 ;; Bit definitions		0x1D	CANSTAT
#define trOPMOD2	CANSTAT, 7	0x1E	CANSTAT, 7
#define trOPMOD1	CANSTAT, 6	0x1F	CANSTAT, 6
#define trOPMOD0	CANSTAT, 5	0x20	CANSTAT, 5
#define trICOD2	CANSTAT, 3	0x21	CANSTAT, 3
#define trICOD1	CANSTAT, 2	0x22	CANSTAT, 2
#define trICOD0	CANSTAT, 1	0x23	CANSTAT, 1
 ;; Bit definitions		0x24	CANCTRL
#define trREQOP2	CANCTRL, 7	0x25	CANCTRL, 7
#define trREQOP1	CANCTRL, 6	0x26	CANCTRL, 6
#define trREQOP0	CANCTRL, 5	0x27	CANCTRL, 5
#define trABAT	CANCTRL, 4	0x28	CANCTRL, 4
#define trCLKEN	CANCTRL, 2	0x29	CANCTRL, 2
#define trCLKPRE1	CANCTRL, 1	0x2A	CANCTRL, 1
#define trCLKPRE0	CANCTRL, 0	0x2B	CANCTRL, 0
 ;; Bit definitions		0x2C	CNF3
#define trWAKFIL	CNF3, 6	0x2D	CNF3, 6
#define trPHSEG22	CNF3, 2	0x2E	CNF3, 2
#define trPHSEG21	CNF3, 1	0x2F	CNF3, 1
#define trPHSEG20	CNF3, 0	0x30	CNF3, 0
 ;; Bit definitions		0x31	CNF2
#define trBTLMODE	CNF2, 7	0x32	CNF2, 7
#define trSAM	CNF2, 6	0x33	CNF2, 6
#define trPHSEG12	CNF2, 5	0x34	CNF2, 5
#define trPHSEG11	CNF2, 4	0x35	CNF2, 4
#define trPHSEG10	CNF2, 3	0x36	CNF2, 3
#define trPHSEG2	CNF2, 2	0x37	CNF2, 2
#define trPHSEG1	CNF2, 1	0x38	CNF2, 1
#define trPHSEG0	CNF2, 0	0x39	CNF2, 0

```

;; Bit definitions      CNF1
#define trSJW1      CNF1,7
#define trSJW0      CNF1,6
#define trBRP5      CNF1,5
#define trBRP4      CNF1,4
#define trBRP3      CNF1,3
#define trBRP2      CNF1,2
#define trBRP1      CNF1,1
#define trBRP0      CNF1,0

;; Bit definitions CANINTE
#define trMERRE      CANINTE,7
#define trWAKIE      CANINTE,6
#define trERRIE      CANINTE,5
#define trTX2IE      CANINTE,4
#define trTX1IE      CANINTE,3
#define trTX0IE      CANINTE,2
#define trRX1IE      CANINTE,1
#define trRX0IE      CANINTE,0

;; Bit definitions CANINTF
#define trMERRF      CANINTF,7
#define trWAKIF      CANINTF,6
#define trERRIF      CANINTF,5
#define trTX2IF      CANINTF,4
#define trTX1IF      CANINTF,3
#define trTX0IF      CANINTF,2
#define trRX1IF      CANINTF,1
#define trRX0IF      CANINTF,0

;; Bit definitions EFLG
#define trRX1OVR    EFLG,7
#define trRX0OVR    EFLG,6
#define trTXB0      EFLG,5
#define trTXEP      EFLG,4
#define trRXEP      EFLG,3
#define trTXWAR    EFLG,2
#define trRXWAR    EFLG,1
#define trEWARN    EFLG,0

;; Bit definitions TXB0CTRL
#define trABTF0     TXB0CTRL,6
#define trMLOA0     TXB0CTRL,5
#define trTXERR0    TXB0CTRL,4
#define trTXREQ0    TXB0CTRL,3
#define trTXP10     TXB0CTRL,1
#define trTXP00     TXB0CTRL,0

;; Bit definitions TXB1CTRL
#define trABTF1     TXB1CTRL,6
#define trMLOA1     TXB1CTRL,5
#define trTXERR1    TXB1CTRL,4
#define trTXREQ1    TXB1CTRL,3
#define trTXP11     TXB1CTRL,1
#define trTXP01     TXB1CTRL,0

;; Bit definitions TXB2CTRL
#define trABTF2     TXB2CTRL,6
#define trMLOA2     TXB2CTRL,5
#define trTXERR2    TXB2CTRL,4
#define trTXREQ2    TXB2CTRL,3
#define trTXP12     TXB2CTRL,1
#define trTXP02     TXB2CTRL,0

```

2 Application Notes

```

;; Bit definitions RXB0CTRL
#define trRXM10    RXB0CTRL,6
#define trRXM00    RXB0CTRL,5
#define trRXRTR0    RXB0CTRL,3
#define trBUKT01    RXB0CTRL,2
#define trBUKT00    RXB0CTRL,1
#define trFILHIT00    RXB0CTRL,0

;; Bit definitions RXB1CTRL
#define trRXM11    RXB1CTRL,6
#define trRXM01    RXB1CTRL,5
#define trRXRTR1    RXB1CTRL,3
#define trFILHIT12    RXB1CTRL,2
#define trFILHIT11    RXB1CTRL,1
#define trFILHIT10    RXB1CTRL,0

;; use with SPI_Rts function
#define RTS0        0x01
#define RTS1        0x02
#define RTS2        0x04

;***** MCP2510 Instructions *****
;***** MCP2510 Read Instruction *****
#define d2510Rd      0x03      ; MCP2510 read instruction
;***** MCP2510 Write Instruction *****
#define d2510Wrt     0x02      ; MCP2510 write instruction
;***** MCP2510 Reset Instruction *****
#define d2510Reset   0xC0      ; MCP2510 reset instruction
;***** MCP2510 RTS Instruction *****
#define d2510RTS     0x80      ; MCP2510 RTS instruction
;***** MCP2510 Status Instruction *****
#define d2510Status   0xA0      ; MCP2510 Status instruction
;***** MCP2510 Bit Modify Instruction *****
#define d2510BitMod  0x05      ; MCP2510 bit modify instruction

```

0,0000	0,0000 0000 0000 0000
0,0001	0,0000 0000 0000 0001
0,0010	0,0000 0000 0000 0010
0,0011	0,0000 0000 0000 0011
0,0100	0,0000 0000 0000 0100
0,0101	0,0000 0000 0000 0101
0,0110	0,0000 0000 0000 0110
0,0111	0,0000 0000 0000 0111
0,1000	0,0000 0000 0000 1000
0,1001	0,0000 0000 0000 1001
0,1010	0,0000 0000 0000 1010
0,1011	0,0000 0000 0000 1011
0,1100	0,0000 0000 0000 1100
0,1101	0,0000 0000 0000 1101
0,1110	0,0000 0000 0000 1110
0,1111	0,0000 0000 0000 1111
1,0000	0,0000 0000 0001 0000
1,0001	0,0000 0000 0001 0001
1,0010	0,0000 0000 0001 0010
1,0011	0,0000 0000 0001 0011
1,0100	0,0000 0000 0001 0100
1,0101	0,0000 0000 0001 0101
1,0110	0,0000 0000 0001 0110
1,0111	0,0000 0000 0001 0111
1,1000	0,0000 0000 0001 1000
1,1001	0,0000 0000 0001 1001
1,1010	0,0000 0000 0001 1010
1,1011	0,0000 0000 0001 1011
1,1100	0,0000 0000 0001 1100
1,1101	0,0000 0000 0001 1101
1,1110	0,0000 0000 0001 1110
1,1111	0,0000 0000 0001 1111
2,0000	0,0000 0000 0010 0000
2,0001	0,0000 0000 0010 0001
2,0010	0,0000 0000 0010 0010
2,0011	0,0000 0000 0010 0011
2,0100	0,0000 0000 0010 0100
2,0101	0,0000 0000 0010 0101
2,0110	0,0000 0000 0010 0110
2,0111	0,0000 0000 0010 0111
2,1000	0,0000 0000 0010 1000
2,1001	0,0000 0000 0010 1001
2,1010	0,0000 0000 0010 1010
2,1011	0,0000 0000 0010 1011
2,1100	0,0000 0000 0010 1100
2,1101	0,0000 0000 0010 1101
2,1110	0,0000 0000 0010 1110
2,1111	0,0000 0000 0010 1111
3,0000	0,0000 0000 0011 0000
3,0001	0,0000 0000 0011 0001
3,0010	0,0000 0000 0011 0010
3,0011	0,0000 0000 0011 0011
3,0100	0,0000 0000 0011 0100
3,0101	0,0000 0000 0011 0101
3,0110	0,0000 0000 0011 0110
3,0111	0,0000 0000 0011 0111
3,1000	0,0000 0000 0011 1000
3,1001	0,0000 0000 0011 1001
3,1010	0,0000 0000 0011 1010
3,1011	0,0000 0000 0011 1011
3,1100	0,0000 0000 0011 1100
3,1101	0,0000 0000 0011 1101
3,1110	0,0000 0000 0011 1110
3,1111	0,0000 0000 0011 1111
4,0000	0,0000 0000 0100 0000
4,0001	0,0000 0000 0100 0001
4,0010	0,0000 0000 0100 0010
4,0011	0,0000 0000 0100 0011
4,0100	0,0000 0000 0100 0100
4,0101	0,0000 0000 0100 0101
4,0110	0,0000 0000 0100 0110
4,0111	0,0000 0000 0100 0111
4,1000	0,0000 0000 0100 1000
4,1001	0,0000 0000 0100 1001
4,1010	0,0000 0000 0100 1010
4,1011	0,0000 0000 0100 1011
4,1100	0,0000 0000 0100 1100
4,1101	0,0000 0000 0100 1101
4,1110	0,0000 0000 0100 1110
4,1111	0,0000 0000 0100 1111
5,0000	0,0000 0000 0101 0000
5,0001	0,0000 0000 0101 0001
5,0010	0,0000 0000 0101 0010
5,0011	0,0000 0000 0101 0011
5,0100	0,0000 0000 0101 0100
5,0101	0,0000 0000 0101 0101
5,0110	0,0000 0000 0101 0110
5,0111	0,0000 0000 0101 0111
5,1000	0,0000 0000 0101 1000
5,1001	0,0000 0000 0101 1001
5,1010	0,0000 0000 0101 1010
5,1011	0,0000 0000 0101 1011
5,1100	0,0000 0000 0101 1100
5,1101	0,0000 0000 0101 1101
5,1110	0,0000 0000 0101 1110
5,1111	0,0000 0000 0101 1111
6,0000	0,0000 0000 0110 0000
6,0001	0,0000 0000 0110 0001
6,0010	0,0000 0000 0110 0010
6,0011	0,0000 0000 0110 0011
6,0100	0,0000 0000 0110 0100
6,0101	0,0000 0000 0110 0101
6,0110	0,0000 0000 0110 0110
6,0111	0,0000 0000 0110 0111
6,1000	0,0000 0000 0110 1000
6,1001	0,0000 0000 0110 1001
6,1010	0,0000 0000 0110 1010
6,1011	0,0000 0000 0110 1011
6,1100	0,0000 0000 0110 1100
6,1101	0,0000 0000 0110 1101
6,1110	0,0000 0000 0110 1110
6,1111	0,0000 0000 0110 1111
7,0000	0,0000 0000 0111 0000
7,0001	0,0000 0000 0111 0001
7,0010	0,0000 0000 0111 0010
7,0011	0,0000 0000 0111 0011
7,0100	0,0000 0000 0111 0100
7,0101	0,0000 0000 0111 0101
7,0110	0,0000 0000 0111 0110
7,0111	0,0000 0000 0111 0111
7,1000	0,0000 0000 0111 1000
7,1001	0,0000 0000 0111 1001
7,1010	0,0000 0000 0111 1010
7,1011	0,0000 0000 0111 1011
7,1100	0,0000 0000 0111 1100
7,1101	0,0000 0000 0111 1101
7,1110	0,0000 0000 0111 1110
7,1111	0,0000 0000 0111 1111

```

;***** SPECIAL CAN MACROS *****
;***** SPI_Read macro Reg
; Read MCP2510 register Reg and return data in W.
;***** SPI_Read macro Reg
    movlw    Reg
    call    Rd2510Reg
    endm

;***** SPI_WriteL macro Reg,LitData
; Write literal byte to MCP2510 register Reg.
;***** SPI_WriteL macro Reg,LitData
    movlw    LitData
    movwf    b2510RegData
    movlw    Reg
    call    Wrt2510Reg
    endm

;***** SPI_WriteV macro Reg,RegData
; Write Data byte to MCP2510 register Reg.
;***** SPI_WriteV macro Reg,RegData
    movfw    RegData
    movwf    b2510RegData
    movlw    Reg
    call    Wrt2510Reg
    endm

;***** SPI_WriteW macro Reg
; Write W byte to MCP2510 register Reg.
;***** SPI_WriteW macro Reg
    movwf    b2510RegData
    movlw    Reg
    call    Wrt2510Reg
    endm

;***** SPI_BitMod macro Reg,Mask,Data
; Write bits determined by Mask & Data to MCP2510 register Reg.
;***** SPI_BitMod macro Reg,Mask,Data
    movlw    Mask
    movwf    b2510RegMask
    movlw    Data
    movwf    b2510RegData
    movlw    Reg
    call    BitMod2510
    endm

;***** SPI_Rts macro Data
; Arm xmit buffers for xmission
;***** SPI_Rts macro Data
    movlw    Data
    call    Rts2510
    endm

```

AN212

```

;*****
;***** Support routines for communicating with MCP2510 chip *****
;*****
;***** CheckCANMsg
;
; Checks for message in Receive Buf 1. If no message pending return
; with Z flag set.
;
; If message pending:
;   Load iRecID_L,iRecID_H with ID.
;   Load bRecCount with number of bytes of data received.
;   Load buffer at pRecDataBase with data
;   Clear MCP2510 Receive Buffer 1 interrupt flag
;   Set tbRxMsgPend flag and clear Z flag.
;
; NOTE: If message already pending doesn't check for new message.
;
;*****
CheckCANMsg

        bcf      _Z          ; for return
        skipClr  tbRxMsgPend ; new CAN message received
        return    ; Message already pending

;; Test for Message pending in Receive Buffer 1
        SPI_Read  CANINTF
        andlw   0x02

        skipNZ
        return    ; Nothing in Rec Buf 1

        bsf      tbRxMsgPend ; new CAN message received

;; Get ID of message source
        SPI_Read  RXB1SIDH
        movwf   iRecID_H
        SPI_Read  RXB1SIDL
        andlw   0xE0
        movwf   iRecID_L

;; Get number of bytes of data
        SPI_Read  RXB1DLC
        andlw   0x0F
        movwf   bRecCount

;; Get data from buffer. Up to 8 bytes based on
        clrf      bCnt

jRxChk11 jmpFeqF  bCnt,bRecCount,jRxChk90           ; no data left

;; Calculate correct MCP2510 receive buffer location
        movlw    RXB1D0
        addwf   bCnt,W

;; Get data byte
        call     Rd2510Reg
        movwf   b2510RegData    ; temporary save

;; Calculate destination buffer location
        movlw    pRecDataBase
        addwf   bCnt,W

```

```
        movwf      FSR
        ; Store data in buffer
        movfw    b2510RegData ; temporary save
        movwf    INDF
        incf    bCnt,F
        goto    jRxChk11
jRxChk90
        SPI_BitMod CANINTF,0x02,0 ; Clear receive buffer 1 ;interrupt
        bcf     _Z
        ; signal data pending
        return
```

di team new USA on 8 mi sandhie mark and i
measures blood of exaggerated hero . dies ed on hand

AN212

```

;*****
;SetConfigMode
;
;// Function Name: Set_Config_Mode()
;*****
SetConfigMode
; SPI_BitMod(CANCTRL, 0xE0, 0x80); //Config. mode/
    bL2bV    0xE0,b2510RegMask
    bL2bV    0x80,b2510RegData
    movlw    CANCTRL
    call     BitMod2510

jSetConfigM1
    movlw    CANSTAT
    call     Rd2510Reg
    andlw   0xE0
    xorlw   0x80
    jmpNZ   jSetConfigM1

    return

;*****
;SetNormalMode
;
;// Function Name: Set_Normal_Mode()
;*****
SetNormalMode
    bL2bV    0xE0,b2510RegMask
    bL2bV    0x00,b2510RegData
    movlw    CANCTRL
    call     BitMod2510

jSetNormalM1
    movlw    CANSTAT
    call     Rd2510Reg
    andlw   0xE0
    jmpNZ   jSetNormalM1

    return

;*****
;WaitANDeqZ
;        Wait for byte from address in W to AND with mask in
;        b2510RegMask to be zero. Uses b2510RegAdr to hold address.
;
;        ;*****
WaitANDeqZ
    movwf   b2510RegAdr      ; save

jWaitANDeqZ
    movfw   b2510RegAdr      ; save
    call    Rd2510Reg
    andwf   b2510RegMask,W
    jmpNZ   jWaitANDeqZ

    return

```

```

; ****
; ***** BASIC COMMUNICATION *****
; ****
; Get2510Status      ; Get Status byte from MCP2510.
; // Function Name: SPI_ReadStatus()
;
Get2510Status
    call    InitSPIBuf
    movlw  d2510Status        ; MCP2510 Status instruction
    call    LoadSPIByte
    movlw  1                  ; expect 1 byte answer
    call    LoadSPIZeros
    call    ExchangeSPI
    call    WaitSPIExchange
    return

; ****
; Rd2510Reg          ; Read MCP2510 register at address in W. Return results
; in W. Uses b2510RegAddr to hold address.
; // Function Name: SPI_Read(uint address)
;
Rd2510Reg
    movwf  b2510RegAddr       ; save
    call    InitSPIBuf
    movlw  d2510RD            ; MCP2510 read instruction
    call    LoadSPIByte
    movfw  b2510RegAddr       ; get address
    call    LoadSPIByte
    movlw  1                  ; expect 1 byte answer
    call    LoadSPIZeros
    call    ExchangeSPI
    call    WaitSPIExchange
    movfw  pSPIBufBase+2
    return

; ****
; Wrt2510Reg          ; Write byte in b2510RegData to MCP2510 register at location in W.
; Uses b2510RegAddr to hold address.
; // Function Name: SPI_Write(uint address)
;
Wrt2510Reg
    movwf  b2510RegAddr       ; save
    call    InitSPIBuf
    movlw  d2510Wrt           ; MCP2510 write instruction
    call    LoadSPIByte
    movfw  b2510RegAddr       ; get address
    call    LoadSPIByte
    movfw  b2510RegData        ; get data
    call    LoadSPIByte
    call    ExchangeSPI
    call    WaitSPIExchange
    return

```

```
;*****  
;BitMod2510  
;// Function Name: SPI_BitMod()  
;      Write data in b2510RegData using mask in b2510RegMask to  
;      address in W. Uses b2510RegAdr to hold address.  
;*****  
BitMod2510  
    movwf    b2510RegAdr      ; save  
    call     InitsPIBuf  
  
    movlw    d2510BitMod      ; MCP2510 bit modify ;instruction  
    call     LoadSPIByte  
  
    movfw    b2510RegAdr      ; address  
    call     LoadSPIByte  
  
    movfw    b2510RegMask      ; mask 01000000  
    call     LoadSPIByte  
  
    movfw    b2510RegData      ; data  
    call     LoadSPIByte  
  
    call     ExchangeSPI  
    call     WaitSPIExchange  
    return  
  
;*****  
;Rts2510  
;      Request to send to MCP2510.  
;      Send the request to send instruction to the CANbus Controller ORed  
;      with value in W. Uses b2510RegData.  
;// Function Name: SPI_Reset()  
;*****  
Rts2510  
    movwf    b2510RegData      ; set value for  
    call     InitsPIBuf  
  
    movlw    d2510RTS          ; MCP2510 RTS instruction  
    iorwf    b2510RegData,W      ; get data and OR it with RTS  
    call     LoadSPIByte  
  
    call     ExchangeSPI  
    call     WaitSPIExchange  
    return  
  
;*****  
;Reset2510  
;      Reset MCP2510.  
;// Function Name: SPI_Reset()  
;*****  
Reset2510  
    call     InitsPIBuf  
    movlw    d2510Reset         ; MCP2510 reset instruction  
    call     LoadSPIByte  
    call     ExchangeSPI  
    call     WaitSPIExchange  
    return
```

```

;***** LOCAL - DON'T CALL DIRECTLY *****
;***** LOCAL - DON'T CALL DIRECTLY *****

;InitSPIPort
;    Initialize SPI port
;InitSPIPort
    BANK0
        bcf    _SSPEN      ; disable SPI
        movlw  0x11      ; SPI Master, Idle high, Fosc/16
        movwf  SSPCON
        bsf    _SSPEN      ; enable SPI
        bcf    _SSPIF      ; clear interrupt flag
    BANK1
        bsf    _SSPIE_P    ; SSP int enable (BANK 1)
    BANK0
        return
;InitSPIBuf
;    Initializes SPI buffer for transaction. Sets up
;    FSR as buffer pointer.
;InitSPIBuf
        clrf   bSPICnt
        movlw  pSPIBufBase
        movwf  pSPIBuf
        movwf  FSR
        return

;LoadSPIByte
;    Load byte in W to SPI buffer. Assumes FSR is pointer.
;LoadSPIByte
        movwf  INDF
        incf   FSR,F
        return

;LoadSPIZeros
;    Load number of zeros in W to SPI buffer.
;    Assumes FSR is pointer.
;LoadSPIZeros
        andlw  0xFF
        skipNZ
        return
;    finished
        clrf   INDF
        incf   FSR,F
        addlw  0xFF      ; Subtract 1 from W
        jmpNZ  LoadSPIZeros
        return

```

```
;*****
;ExchangeSPI
;    Initiate SPI transaction.
;*****
ExchangeSPI
    ; Get number of bytes to exchange
    bV2bV    FSR,bSPICnt
    movlw    pSPIBufBase
    subwf    bSPICnt,F

    skipNZ
    return      ; nothing to exchange

    movlw    pSPIBufBase
    movwf    pSPIBuf

    ; Load 1st byte to begin exchange
    bcf      tp2510_CS_          ; CS_ for MCP2510 chip
    movfw    pSPIBufBase        ; get 1st byte in buffer
    movwf    SSPBUF              ; send it
    return

;*****
;WaitSPIExchange
;    Wait for SPI transaction to be completed.
;*****
WaitSPIExchange
    jmpFneZ  bSPICnt,WaitSPIExchange
    return

;*****
; MCP2510 Instructions
#define d2510Rd    0x03    ; MCP2510 read instruction
#define d2510Wrt   0x02    ; MCP2510 write instruction
#define d2510Reset  0xC0    ; MCP2510 reset instruction
#define d2510RTS   0x80    ; MCP2510 RTS instruction
#define d2510Status 0xA0    ; MCP2510 Status instruction
#define d2510BitMod 0x05    ; MCP2510 bit modify instruction
```

```

;***** SPECIAL CAN MACROS *****
; Read MCP2510 register Reg and return data in W.
SPI_Read macro Reg
    movlw    Reg
    call     Rd2510Reg
endm

; Write literal byte to MCP2510 register Reg.
SPI_WriteL macro Reg,LitData
    movlw    LitData
    movwf    b2510RegData
    movlw    Reg
    call     Wrt2510Reg
endm

; Write Data byte to MCP2510 register Reg.
SPI_WriteV macro Reg,RegData
    movfw    RegData
    movwf    b2510RegData
    movlw    Reg
    call     Wrt2510Reg
endm

; Write W byte to MCP2510 register Reg.
SPI_WriteW macro Reg
    movwf    b2510RegData
    movlw    Reg
    call     Wrt2510Reg
endm

; Write bits determined by Mask & Data to MCP2510 register Reg.
SPI_BitMod macro Reg,Mask,Data
    movlw    Mask
    movwf    b2510RegMask
    movlw    Data
    movwf    b2510RegData
    movlw    Reg
    call     BitMod2510
endm

; Arm xmit buffers for xmission
SPI_Rts macro Data
    movlw    Data
    call     Rts2510
endm

```

AMICO Support routine

```

;support routines for communicating with MCP2510 chip
;*****RECEIVE BUFFER 1***** ; receive buffer 1
;*****RECEIVE BUFFER 2***** ; receive buffer 2
;*****RECEIVE BUFFER 3***** ; receive buffer 3

;CheckCANMsg
;
; Checks for message in Receive Buf 1. If no message pending return
; with Z flag set.
;
; If message pending:
;   Load iRecID_L,iRecID_H with ID.
;   Load bRecCount with number of bytes of data received.
;   Load buffer at pRecDataBase with data
;   Clear MCP2510 Receive Buffer 1 interrupt flag
;   Set tbRxMsgPend flag and clear Z flag.
;
; NOTE: If message already pending doesn't check for new message.
;
;*****RECEIVE BUFFER 1***** ; receive buffer 1
;*****RECEIVE BUFFER 2***** ; receive buffer 2
;*****RECEIVE BUFFER 3***** ; receive buffer 3

CheckCANMsg
        bcf      _Z          ; for return
        skipClr tbRxMsgPend ; new CAN message received
        return      ; Message already pending

; Test for Message pending in Receive Buffer 1
        SPI_Read  CANINTF
        andlw    0x02

        skipNZ
        return      ; Nothing in Rec Buf 1

        bsf      tbRxMsgPend ; new CAN message received

; Get ID of message source
        SPI_Read  RXB1SIDH
        movwf    iRecID_H
        SPI_Read  RXB1SIDL
        andlw    0xE0
        movwf    iRecID_L

; Get number of bytes of data
        SPI_Read  RXB1DLC
        andlw    0XF
        movwf    bRecCount

; Get data from buffer. Up to 8 bytes based on
        clrf      bCnt

jRxChk11 jmpFeqF  bCnt,bRecCount,jRxChk90           ; no data left

; Calculate correct MCP2510 receive buffer location
        movlw    RXB1D0
        addwf    bCnt,W

; Get data byte
        call     Rd2510Reg
        movwf    b2510RegData ; temporary save

; Calculate destination buffer location
        movlw    pRecDataBase
        addwf    bCnt,W

```

```
        movwf    FSR
        ;; Store data in buffer
        movfw    b2510RegData      ; temporary save
        movwf    INDF
        incf    bCnt,F
        goto    jRxChk11
jRxChk90
        SPI_BitMod CANINTF,0x02,0
        bcf    _Z
        ; Clear receive buffer 1 ;interrupt
        ; signal data pending
        return
```

at least one bit of the receive mask must be set to 1. This
means the bit of interest must be set to 1.

```
;*****  
;SetConfigMode  
;  
;// Function Name: Set_Config_Mode()  
;*****  
SetConfigMode  
; SPI_BitMod(CANCTRL, 0xE0, 0x80);      //Config. mode/  
    bL2bV    0xE0,b2510RegMask  
    bL2bV    0x80,b2510RegData  
    movlw    CANCTRL  
    call     BitMod2510  
  
jSetConfigM1  
    movlw    CANSTAT  
    call     Rd2510Reg  
    andlw   0xE0  
    xorlw   0x80  
    jmpNZ   jSetConfigM1  
  
    return  
  
;  
;*****  
;SetNormalMode  
;  
;// Function Name: Set_Normal_Mode()  
;*****  
SetNormalMode  
  
    bL2bV    0xE0,b2510RegMask  
    bL2bV    0x00,b2510RegData  
    movlw    CANCTRL  
    call     BitMod2510  
  
jSetNormalM1  
    movlw    CANSTAT  
    call     Rd2510Reg  
    andlw   0xE0  
    jmpNZ   jSetNormalM1  
  
    return  
  
;  
;*****  
;WaitANDeqZ  
;  
;      Wait for byte from address in W to AND with mask in  
;      b2510RegMask to be zero. Uses b2510RegAdr to hold address.  
;  
;*****  
WaitANDeqZ  
    movwf   b2510RegAdr      ; save  
  
jWaitANDeqZ  
    movfw   b2510RegAdr      ; save  
    call    Rd2510Reg  
    andwf   b2510RegMask,W  
    jmpNZ   jWaitANDeqZ  
    return
```

```

; ****
; ****
; **** BASIC COMMUNICATION ****
; ****
; ****
; Get2510Status      ; Read MCP2510 Status byte from MCP2510.
; // Function Name: SPI_ReadStatus()
; ****
Get2510Status
    call    InitSPIBuf
    movlw  d2510Status        ; MCP2510 Status instruction
    call    LoadSPIByte
    movlw  1                  ; expect 1 byte answer
    call    LoadSPIZeros
    call    ExchangeSPI
    call    WaitSPIExchange
    return
; ****
;Rd2510Reg
;     Read MCP2510 register at address in W. Return results
;     in W. Uses b2510RegAddr to hold address.
; // Function Name: SPI_Read(uint address)
; ****
Rd2510Reg
    movwf  b2510RegAddr       ; save
    call    InitSPIBuf
    movlw  d2510Rd            ; MCP2510 read instruction
    call    LoadSPIByte
    movfw  b2510RegAddr       ; get address
    call    LoadSPIByte
    movlw  1                  ; expect 1 byte answer
    call    LoadSPIZeros
    call    ExchangeSPI
    call    WaitSPIExchange
    movfw  pSPIBufBase+2
    return
; ****
;Wrt2510Reg
;     Write byte in b2510RegData to MCP2510 register at location in W.
;     Uses b2510RegAddr to hold address.
; // Function Name: SPI_Write(uint address)
; ****
Wrt2510Reg
    movwf  b2510RegAddr       ; save
    call    InitSPIBuf
    movlw  d2510Wrt           ; MCP2510 write instruction
    call    LoadSPIByte
    movfw  b2510RegAddr       ; get address
    call    LoadSPIByte
    movfw  b2510RegData       ; get data
    call    LoadSPIByte
    call    ExchangeSPI
    call    WaitSPIExchange
    return

```

```
;*****  
;BitMod2510  
// Function Name: SPI_BitMod()  
;      Write data in b2510RegData using mask in b2510RegMask to  
;      address in W. Uses b2510RegAdr to hold address.  
;*****  
BitMod2510  
    movwf    b2510RegAdr      ; save  
    call     InitSPIBuf  
  
    movlw    d2510BitMod      ; MCP2510 bit modify ;instruction  
    call     LoadSPIByte  
  
    movfw    b2510RegAdr      ; address  
    call     LoadSPIByte  
  
    movfw    b2510RegMask      ; mask  
    call     LoadSPIByte  
  
    movfw    b2510RegData      ; data  
    call     LoadSPIByte  
  
    call     ExchangeSPI  
    call     WaitSPIExchange  
    return  
  
;*****  
;Rts2510  
// Function Name: SPI_Reset()  
;      Request to send to MCP2510.  
;      Send the request to send instruction to the CANbus Controller ORed  
;      with value in W. Uses b2510RegData.  
;*****  
Rts2510  
    movwf    b2510RegData      ; save data  
    call     InitSPIBuf  
  
    movlw    d2510RTS         ; MCP2510 RTS instruction  
    iorwf    b2510RegData,W    ; get data and OR it with RTS  
    call     LoadSPIByte  
  
    call     ExchangeSPI  
    call     WaitSPIExchange  
    return  
  
;*****  
;Reset2510  
// Function Name: SPI_Reset()  
;      Reset MCP2510.  
;*****  
Reset2510  
    call     InitSPIBuf  
    movlw    d2510Reset        ; MCP2510 reset instruction  
    call     LoadSPIByte  
    call     ExchangeSPI  
    call     WaitSPIExchange  
    return
```

```

;*****
;***** LOCAL - DON'T CALL DIRECTLY *****
;*****

;***** Initialize SPI port *****
;***** Local - don't call directly *****
;***** 9.000000 *****
InitSPIPort
BANK0
    bcf    _SSPEN      ; disable SPI
    movlw  0x11      ; SPI Master, Idle high, Fosc/16
    movwf  SSPCON
    bsf    _SSPEN      ; enable SPI
    bcf    _SSPIF      ; clear interrupt flag
BANK1
    bsf    _SSPIE_P    ; SSP int enable (BANK 1)
BANK0
    return

;***** Initialize SPI buffer *****
;***** Local - don't call directly *****
;***** 9.000000 *****
InitSPIBuf
;      Initializes SPI buffer for transaction. Sets up
;      FSR as buffer pointer.
;***** Local - don't call directly *****
InitSPIBuf
    clrf   bSPICnt
    movlw  pSPIBufBase
    movwf  pSPIBuf
    movwf  FSR
    return

;***** Load SPI Byte *****
;***** Local - don't call directly *****
;***** 9.000000 *****
LoadSPIByte
;      Load byte in W to SPI buffer. Assumes FSR is pointer.
;***** Local - don't call directly *****
LoadSPIByte
    movwf  INDF
    incf   FSR,F
    return

;***** Load SPI Zeros *****
;***** Local - don't call directly *****
;***** 9.000000 *****
LoadSPIZeros
;      Load number of zeros in W to SPI buffer.
;      Assumes FSR is pointer.
;***** Local - don't call directly *****
LoadSPIZeros
    andlw  0xFF
    skipNZ
    return
    clrf   INDF
    incf   FSR,F
    addlw  0xFF      ; Subtract 1 from W
    jmpNZ  LoadSPIZeros
    return

```

```
;*****  
;ExchangeSPI  
;     Initiate SPI transaction.  
;*****  
ExchangeSPI  
    ;; Get number of bytes to exchange  
    b2bV      FSR,bSPICnt  
    movlw    pSPIBufBase  
    subwf    bSPICnt,F  
  
    skipNZ  
    return      142 addsub ; nothing to exchange  
    movlw    pSPIBufBase  
    movwf    pSPIBuf  
    142 addsub  
    addsub addsub addsub  
  
    ;; Load 1st byte to begin exchange  
    bcf      tp2510_CS_ ; CS_ for MCP2510 chip  
    movfw    pSPIBufBase ; get 1st byte in buffer  
    movwf    SSPBUF ; send it  
    return  
  
;WaitSPIExchange  
;     Wait for SPI transaction to be completed.  
;*****  
WaitSPIExchange  
    jmpfneZ  bSPICnt,WaitSPIExchange  
    return  
  
;
```

```

;-----+
;MCP2510.INC
; Description: This file contains the definitions for the MicroChip
; standalone CANbus controller.
;
; 07/17/99 JPF Original Version
; 09/11/99 JCT Modified for ASM
;-----+



#define RXF0SIDH 0x00
#define RXF0SIDL 0x01
#define RXF0EID8 0x02
#define RXF0EID0 0x03
#define RXF1SIDH 0x04
#define RXF1SIDL 0x05
#define RXF1EID8 0x06
#define RXF1EID0 0x07
#define RXF2SIDH 0x08
#define RXF2SIDL 0x09
#define RXF2EID8 0x0A
#define RXF2EID0 0x0B
#define BFPCTRL 0x0C
#define TXRTSCTRL 0x0D
#define CANSTAT 0x0E
#define CANCTRL 0x0F

#define RXF3SIDH 0x10
#define RXF3SIDL 0x11
#define RXF3EID8 0x12
#define RXF3EID0 0x13
#define RXF4SIDH 0x14
#define RXF4SIDL 0x15
#define RXF4EID8 0x16
#define RXF4EID0 0x17
#define RXF5SIDH 0x18
#define RXF5SIDL 0x19
#define RXF5EID8 0x1A
#define RXF5EID0 0x1B
#define TEC 0x1C
#define REC 0x1D
#define CANSTAT1 0x1E
#define CANCTRL1 0x1F

#define RXM0SIDH 0x20
#define RXM0SIDL 0x21
#define RXM0EID8 0x22
#define RXM0EID0 0x23
#define RXM1SIDH 0x24
#define RXM1SIDL 0x25
#define RXM1EID8 0x26
#define RXM1EID0 0x27
#define CNF3 0x28
#define CNF2 0x29
#define CNF1 0x2A
#define CANINTE 0x2B
#define CANINTF 0x2C
#define EFLG 0x2D
#define CANSTAT2 0x2E
#define CANCTRL2 0x2F

#define TXB0CTRL 0x30
#define TXB0SIDH 0x31
#define TXB0SIDL 0x32
#define TXB0EID8 0x33
#define TXB0EID0 0x34

```

#define TXB0DU	0x30	0x00	TXB0DU
#define TXB0D1	0x37	0x00	TXB0D1
#define TXB0D2	0x38	0x00	TXB0D2
#define TXB0D3	0x39	0x00	TXB0D3
#define TXB0D4	0x3A	0x00	TXB0D4
#define TXB0D5	0x3B	0x00	TXB0D5
#define TXB0D6	0x3C	0x00	TXB0D6
#define TXB0D7	0x3D	0x00	TXB0D7
#define CANSTAT3	0x3E	0x00	CANSTAT3
#define CANCTRL3	0x3F	0x00	CANCTRL3
#define TXB1CTRL	0x40	0x00	TXB1CTRL
#define TXB1SIDH	0x41	0x00	TXB1SIDH
#define TXB1SIDL	0x42	0x00	TXB1SIDL
#define TXB1EID8	0x43	0x00	TXB1EID8
#define TXB1EID0	0x44	0x00	TXB1EID0
#define TXB1DLC	0x45	0x00	TXB1DLC
#define TXB1D0	0x46	0x00	TXB1D0
#define TXB1D1	0x47	0x00	TXB1D1
#define TXB1D2	0x48	0x00	TXB1D2
#define TXB1D3	0x49	0x00	TXB1D3
#define TXB1D4	0x4A	0x00	TXB1D4
#define TXB1D5	0x4B	0x00	TXB1D5
#define TXB1D6	0x4C	0x00	TXB1D6
#define TXB1D7	0x4D	0x00	TXB1D7
#define CANSTAT4	0x4E	0x00	CANSTAT4
#define CANCTRL4	0x4F	0x00	CANCTRL4
#define TXB2CTRL	0x50	0x00	TXB2CTRL
#define TXB2SIDH	0x51	0x00	TXB2SIDH
#define TXB2SIDL	0x52	0x00	TXB2SIDL
#define TXB2EID8	0x53	0x00	TXB2EID8
#define TXB2EID0	0x54	0x00	TXB2EID0
#define TXB2DLC	0x55	0x00	TXB2DLC
#define TXB2D0	0x56	0x00	TXB2D0
#define TXB2D1	0x57	0x00	TXB2D1
#define TXB2D2	0x58	0x00	TXB2D2
#define TXB2D3	0x59	0x00	TXB2D3
#define TXB2D4	0x5A	0x00	TXB2D4
#define TXB2D5	0x5B	0x00	TXB2D5
#define TXB2D6	0x5C	0x00	TXB2D6
#define TXB2D7	0x5D	0x00	TXB2D7
#define CANSTAT5	0x5E	0x00	CANSTAT5
#define CANCTRL5	0x5F	0x00	CANCTRL5
#define RXB0CTRL	0x60	0x00	RXB0CTRL
#define RXB0SIDH	0x61	0x00	RXB0SIDH
#define RXB0SIDL	0x62	0x00	RXB0SIDL
#define RXB0EID8	0x63	0x00	RXB0EID8
#define RXB0EID0	0x64	0x00	RXB0EID0
#define RXB0DLC	0x65	0x00	RXB0DLC
#define RXB0D0	0x66	0x00	RXB0D0
#define RXB0D1	0x67	0x00	RXB0D1
#define RXB0D2	0x68	0x00	RXB0D2
#define RXB0D3	0x69	0x00	RXB0D3
#define RXB0D4	0x6A	0x00	RXB0D4
#define RXB0D5	0x6B	0x00	RXB0D5
#define RXB0D6	0x6C	0x00	RXB0D6
#define RXB0D7	0x6D	0x00	RXB0D7
#define CANSTAT6	0x6E	0x00	CANSTAT6
#define CANCTRL6	0x6F	0x00	CANCTRL6
#define RXB1CTRL	0x70	0x00	RXB1CTRL
#define RXB1SIDH	0x71	0x00	RXB1SIDH

```

#define RXB1SIDL      0x72
#define RXB1EID8      0x73
#define RXB1EID0      0x74
#define RXB1DLC       0x75
#define RXB1D0        0x76
#define RXB1D1        0x77
#define RXB1D2        0x78
#define RXB1D3        0x79
#define RXB1D4        0x7A
#define RXB1D5        0x7B
#define RXB1D6        0x7C
#define RXB1D7        0x7D
#define CANSTAT7      0x7E
#define CANCTRL7       0x7F

; Bit definitions

; Bit definitions BFPCTRL
#define trB1BFS        BFPCTRL, 5
#define trB0BFS        BFPCTRL, 4
#define trB1BFE        BFPCTRL, 3
#define trB0BFE        BFPCTRL, 2
#define trB1BFM        BFPCTRL, 1
#define trB0BFM        BFPCTRL, 0

; Bit definitions TXRTSCTRL
#define trB2RTS        BFPCTRL, 5
#define trB1RTS        BFPCTRL, 4
#define trB0RTS        BFPCTRL, 3
#define trB2RTSM       BFPCTRL, 2
#define trB1RTSM       BFPCTRL, 1
#define trB0RTSM       BFPCTRL, 0

; Bit definitions CANSTAT
#define trOPMOD2       CANSTAT, 7
#define trOPMOD1       CANSTAT, 6
#define trOPMOD0       CANSTAT, 5
#define trICOD2        CANSTAT, 3
#define trICOD1        CANSTAT, 2
#define trICOD0        CANSTAT, 1

; Bit definitions CANCTRL
#define trREQOP2       CANCTRL, 7
#define trREQOP1       CANCTRL, 6
#define trREQOP0       CANCTRL, 5
#define trABAT         CANCTRL, 4
#define trCLKEN        CANCTRL, 2
#define trCLKPRE1      CANCTRL, 1
#define trCLKPRE0      CANCTRL, 0

; Bit definitions CNF3
#define trWAKFIL       CNF3, 6
#define trPHSEG22      CNF3, 2
#define trPHSEG21      CNF3, 1
#define trPHSEG20      CNF3, 0

; Bit definitions CNF2
#define trBTLMODE      CNF2, 7
#define trSAM          CNF2, 6
#define trPHSEG12      CNF2, 5
#define trPHSEG11      CNF2, 4
#define trPHSEG10      CNF2, 3
#define trPHSEG2        CNF2, 2
#define trPHSEG1        CNF2, 1
#define trPHSEG0        CNF2, 0

```

```
; Bit definitions CNF1
#define trSJW1      CNF1,7
#define trSJW0      CNF1,6
#define trBRP5      CNF1,5
#define trBRP4      CNF1,4
#define trBRP3      CNF1,3
#define trBRP2      CNF1,2
#define trBRP1      CNF1,1
#define trBRP0      CNF1,0

; Bit definitions CANINTE
#define trMERRE      CANINTE,7
#define trWAKIE      CANINTE,6
#define trERRIE      CANINTE,5
#define trTX2IE      CANINTE,4
#define trTX1IE      CANINTE,3
#define trRX0IE      CANINTE,2
#define trRX1IE      CANINTE,1
#define trRX0IE      CANINTE,0

; Bit definitions CANINTF
#define trMERRF      CANINTF,7
#define trWAKIF      CANINTF,6
#define trERRIF      CANINTF,5
#define trTX2IF      CANINTF,4
#define trTX1IF      CANINTF,3
#define trRX0IF      CANINTF,2
#define trRX1IF      CANINTF,1
#define trRX0IF      CANINTF,0

; Bit definitions EFLG
#define trRX1OVR     EFLG,7
#define trRX0OVR     EFLG,6
#define trTXB0        EFLG,5
#define trTXEP        EFLG,4
#define trRXEP        EFLG,3
#define trTXWAR      EFLG,2
#define trRXWAR      EFLG,1
#define trREWARN     EFLG,0

; Bit definitions TXB0CTRL
#define trABTF0      TXB0CTRL,6
#define trMLOA0      TXB0CTRL,5
#define trTXERR0      TXB0CTRL,4
#define trTXREQ0      TXB0CTRL,3
#define trTXP10      TXB0CTRL,1
#define trTXP00      TXB0CTRL,0

; Bit definitions TXB1CTRL
#define trABTF1      TXB1CTRL,6
#define trMLOA1      TXB1CTRL,5
#define trTXERR1      TXB1CTRL,4
#define trTXREQ1      TXB1CTRL,3
#define trTXP11      TXB1CTRL,1
#define trTXP01      TXB1CTRL,0

; Bit definitions TXB2CTRL
#define trABTF2      TXB2CTRL,6
#define trMLOA2      TXB2CTRL,5
#define trTXERR2      TXB2CTRL,4
#define trTXREQ2      TXB2CTRL,3
#define trTXP12      TXB2CTRL,1
#define trTXP02      TXB2CTRL,0
```



```
;Basic macros for PIC16C series
;6/20/98

#ifndef __16C77
#define _COMMONBANK ; use common upper 16 bytes in 4 banks
#endif
#ifndef __16C76
#define _COMMONBANK ; use common upper 16 bytes in 4 banks
#endif

TRUE      equ    1
FALSE     equ    0

; Page 1 register definitions to avoid page warning

OPTION_REG_P      EQU    H'0081'-0x80
TRISA_P          EQU    H'0085'-0x80
TRISB_P          EQU    H'0086'-0x80
TRISC_P          EQU    H'0087'-0x80
TRISD_P          EQU    H'0088'-0x80
TRISE_P          EQU    H'0089'-0x80
PIE1_P           EQU    H'008C'-0x80
PIE2_P           EQU    H'008D'-0x80
PCON_P           EQU    H'008E'-0x80
PR2_P            EQU    H'0092'-0x80
SSPADD_P         EQU    H'0093'-0x80
SSPSTAT_P        EQU    H'0094'-0x80
TXSTA_P          EQU    H'0098'-0x80
SPBRG_P          EQU    H'0099'-0x80
ADCON1_P         EQU    H'009F'-0x80

; Special register bit definition pairs

; STATUS bit definitions

#define _C           STATUS,0
#define _DC          STATUS,1
#define _Z           STATUS,2
#define _PD          STATUS,3
#define _TO          STATUS,4
#define _RP0         STATUS,5
#define _RP1         STATUS,6
#define _IRP         STATUS,7

#define _INTE        INTCON,INTE ; External interrupt enable
#define _INTF        INTCON,INTF ; External interrupt flag
#define _RBIE        INTCON,RBIE ; Port B pins 4-7 edge interrupt enable
#define _RBIF        INTCON,RBIF ; Port B pins 4-7 edge interrupt flag
#define _TOIE        INTCON,TOIE ; Timer 0 interrupt enable
#define _TOIF        INTCON,TOIF ; Timer 0 interrupt flag

#define _CCP1IE_P    PIE1_P,CCP1IE ; Timer 1 compare int enable (page 1)
#define _CCP1IF      PIR1,CCP1IF ; Timer 1 compare int flag

#define _RCIE_P      PIE1_P,RCIE ; async rec interrupt enable (page 1)
#define _RCIF         PIR1,RCIF ; async rec interrupt flag

#define _TXIE_P      PIE1_P,TXIE ; async xmit interrupt enable (page
;1)
#define _TXIF         PIR1,TXIF ; async xmit interrupt flag

#define _SSPIE_P     PIE1_P,SSPIE ; SSP int enable (page 1)
```

```

#define _SSPIF      PIR1,SSPIE      ; SSP interrupt flag
#define _TMR1IE_P   PIE1_P,TMR1IE ; Timer 1 enable (page 1)
#define _TMR1IF     PIR1,TMR1IF   ; Timer1 interrupt flag

#define _TMR2IE_P   PIE1_P,TMR2IE ; Timer 2 enable (page 1)
#define _TMR2IF     PIR1,TMR2IF   ; Timer2 interrupt flag

#ifndef _COMMONBANK ; use common upper 16 bytes in 4 banks

PAGE3 macro
  bsf    PCLATH,4
  bsf    PCLATH,3
endm

PAGE2 macro
  bsf    PCLATH,4
  bcf    PCLATH,3
endm

PAGE1 macro
  bcf    PCLATH,4
  bsf    PCLATH,3
endm

PAGE0 macro
  bcf    PCLATH,4
  bcf    PCLATH,3
endm

BANK3 macro
  bsf    STATUS,6
  bsf    STATUS,5
endm

BANK2 macro
  bsf    STATUS,6
  bcf    STATUS,5
endm

BANK1 macro
  bcf    STATUS,6
  bsf    STATUS,5
endm

BANK0 macro
  bcf    STATUS,6
  bcf    STATUS,5
endm

FSRBank23 macro
  bsf    STATUS,7
endm

FSRBank01 macro
  bcf    STATUS,7
endm

#else

PAGE1 macro
  bsf    PCLATH,3
endm

```

```

PAGE0 macro          ; Page constraint code : $1420,INIT      page enable
  bcf    PCLATH,3           (1 opnd) address & result : ARITHM,4,CODE      R/W,ENBL,enable
  endm

BANK0 macro          ; (1 opnd) address & result : ARITHM,4,CODE      R/W,ENBL,enable
  bcf    STATUS,5           ; Select page 0 is automatic result : ARITHM,4,CODE      R/W,ENBL,enable
  endm

BANK1 macro          ; (1 opnd) address & result : ARITHM,4,CODE      R/W,ENBL,enable
  bsf    STATUS,5           ; Select page 1
  endm
#endif

enableInt macro
  bsf    INTCON,GIE
  endm

disableInt macro
  local   Loop
Loop  bcf    INTCON,GIE
      btfsc  INTCON,GIE
      goto   Loop
  endm

; Byte logical & arithmetic macros

bV2bV macro  bSource,bDest
  movf   bSource,W
  movwf  bDest
  endm

bL2bV macro  bVal,bDest
  movlw  bVal
  movwf  bDest
  endm

jmpFeqZ macro  Reg,Label
  movf   Reg,F
  btfsc  _Z
  goto   Label
  endm

jmpFneZ macro  Reg,Label
  movf   Reg,F
  btfss  _Z
  goto   Label
  endm

jmpFgtL macro  Reg1,bVal,Label
  movfw  Reg1
  jmpWgtL bVal,Label
  endm

jmpFgeL macro  Reg1,bVal,Label
  movfw  Reg1
  jmpWgeL bVal,Label
  endm

jmpFeqL macro  Reg,bVal,Label
  movf   Reg,W
  sublw  bVal
  btfsc  _Z

```

```

        goto    Label
        endm

jmpFneL macro   Reg,bVal,Label
        movf    Reg,W
        sublw   bVal
        btfss   _Z
        goto    Label
        endm

jmpFleL macro   Reg1,bVal,Label
        movfw   Reg1
        jmpWleL bVal,Label
        endm

jmpFltL macro   Reg1,bVal,Label
        movfw   Reg1
        jmpWltL bVal,Label
        endm

jmpFeeF macro   Reg1,Reg2,Label
        movf    Reg1,W
        subwf   Reg2,W
        btfsc   _Z
        goto    Label
        endm

jmpFneF macro   Reg1,Reg2,Label
        movf    Reg1,W
        subwf   Reg2,W
        btfss   _Z
        goto    Label
        endm

jmpFleF macro   Reg1,Reg2,Label
        movfw   Reg1
        jmpWleF Reg2,Label
        endm

jmpFltF macro   Reg1,Reg2,Label
        movfw   Reg1
        jmpWltF Reg2,Label
        endm

jmpWeqZ macro   Label      ; jmp if W == 0
        andlw  0xFF
        jmpZ   Label
        endm

jmpWneZ macro   Label      ; jmp if W != 0
        andlw  0xFF
        jmpNZ  Label
        endm

skipFeeZ macro   Reg
        movf    Reg,F
        btfss   _Z
        endm

skipFneZ macro   Reg
        movf    Reg,F
        btfsc   _Z
        endm

skipFeeL macro   Reg,bVal

```

```

        sublw    bVal
        btfss    _Z
        endm

skipFneL macro  Reg,bVal
        movf     Reg,W
        sublw    bVal
        btfsc    _Z
        endm

skipFeeF macro  Reg1,Reg2
        movf     Reg1,W
        subwf    Reg2,W
        btfss    _Z
        endm

skipFneF macro  Reg1,Reg2
        movf     Reg1,W
        subwf    Reg2,W
        btfsc    _Z
        endm

skipWeqZ macro
        andlw    0xFF
        btfss    _Z
        endm

skipWneZ macro
        andlw    0xFF
        btfsc    _Z
        endm

jmpWgtL macro  bVal,Label
        sublw    bVal
        btfss    _C
        goto    Label
        endm

jmpWgeL macro  bVal,Label
        sublw    bVal
        btfss    _C
        goto    Label
        btfsc    _Z
        goto    Label
        endm

jmpWeqL macro  bVal,Label
        sublw    bVal
        btfsc    _Z
        goto    Label
        endm

jmpWneL macro  bVal,Label
        sublw    bVal
        btfss    _Z
        goto    Label
        endm

jmpWleL macro  bVal,Label
        sublw    bVal
        btfsc    _C
        goto    Label
        endm

```

Copyright © 2000 Microchip Technology Inc.

```

jmpWltL macro bVal,Label
    sublw bVal
    skipC
    bsf _Z
jmpNZLabel
endm

jmpWgtF macro Reg,Label
    subwf Reg,W
    btfss _C
    goto Label
endm

jmpWgeF macro Reg,Label
    subwf Reg,W
    btfss _C
    goto Label
    btfsc _Z
    goto Label
endm

jmpWeqF macro Reg,Label
    subwf Reg,W
    btfsc _Z
    goto Label
endm

jmpWneF macro Reg,Label
    subwf Reg,W
    btfsc _Z
    goto Label
endm

jmpWleF macro Reg,Label
    subwf Reg,W
    btfsc _C
    goto Label
endm

jmpWltF macro Reg,Label
    subwf Reg,W
    skipC
    bsf _Z
jmpNZ Label
endm

jmpClr macro Reg,Bit,Label
    btfss Reg,Bit
    goto Label
endm

jmpSet macro Reg,Bit,Label
    btfsc Reg,Bit
    goto Label
endm

jmpZ macro Label
    btfsc _Z
    goto Label
endm

jmpNZ macro Label
    btfss _Z

```

AN212

```
        goto    Label
        endm

jmpC    macro Label
        btfsc   _C
        goto    Label
        endm

jmpNC   macro Label
        btfss   _C
        goto    Label
        endm

skipClr macro Reg,Bit
        btfsc   Reg, Bit
        endm

skipSet  macro Reg,Bit
        btfss   Reg, Bit
        endm

skipNZ   macro
        btfsc   _Z
        endm

skipZ    macro
        btfss   _Z
        endm

skipNC   macro
        btfsc   _C
        endm

skipC    macro
        btfss   _C
        endm

toggle   macro Reg,Bit
        local   SLabel,Label
        btfss   Reg, Bit
        goto    SLabel
        bcf    Reg, Bit
        goto    Label
        SLabel bsf   Reg, Bit
        Label
        endm

tb2tb   macro RegS,Bits,RegD,BitD
        local   jLab1,jLab2
        jmpSet RegS,Bits,jLab1
        bcf    RegD,BitD
        goto    jLab2
        jLab1  bsf   RegD,BitD
        jLab2
        endm

tb2Nottb macro RegS,Bits,RegD,BitD
        local   jLab1,jLab2
        jmpClr RegS,Bits,jLab1
        bcf    RegD,BitD
        goto    jLab2
        jLab1  bsf   RegD,BitD
        jLab2
        endm
```

```
ledn,Iav  orca 11Digit
        n  width
        gixa
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv  orca 11Digit
        n  width
        gixa
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc

ledn,Inv,Fwd  orca 11Digit
        n  width
        pcfwd
        l  lsi
        mte 11Digit
        mbsc
```

```

;*****
;Read3201
;      This functions reads MCP3201 and store the result
;      in iA2DValue as a 12 bit value.
;*****
;Read3201

        bcf      tpA2D_CS_           ; CS_ for 3201 A2D chip

        call     InitSPIBuf
        movlw    2                  ; expect 2 bytes
        call     LoadSPIZeros

        ;; Initiate SPI transaction.
        ;; Get number of bytes to exchange
        bv2bV   FSR,bSPICnt
        movlw    pSPIBufBase
        subwf   bSPICnt,F

        movlw    pSPIBufBase
        movwf   pSPIBuf

        ;; Load 1st byte to begin exchange
        movfw   pSPIBufBase          ; get 1st byte in buffer
        movwf   SSPBUF               ; send it

        call     WaitSPIExchange

        bsf      tpA2D_CS_           ; CS_ for 3201 A2D chip

        bv2bV   pSPIBufBase,iA2DValue+1
        bv2bV   pSPIBufBase+1,iA2DValue

        ;; Shift right by 1 to remove extra b1 bit
        iShiftR  iA2DValue

        ;; remove dummy upper 4 bits
        movlw    0x0F
        andwf   iA2DValue+1,F
        return

```

AN212

NOTES:

Temperature Sensing Technologies

Author: Bonnie Baker
Microchip Technology Inc.

INTRODUCTION

Of all of the sensing technologies, temperature sensing is the most common. This phenomena can be explained by citing examples in a multitude of applications where knowing and using the actual or relative temperature is critical. For instance, other sensors such as pressure, force, flow, level, and position many times require temperature monitoring in order to insure accuracy. As an example, pressure and force are usually sensed with resistive Wheatstone bridge configurations. The temperature errors of the resistive elements of these bridges can exceed the actual measurement range of the sensor, making the pressure sensor's output fairly useless, unless the temperature of the bridge is known. Flow and level sensor accuracies are dependent on the density of the liquid or gas.

One variable that affects the accuracy of these sensors is the temperature of that material. Position is most typically used in motor control. In these circuits, temperature affects the efficiency of the motor. Consequently, the understanding of temperature sensing is needed in order to fully understand how to accurately sense most other physical phenomena.

This application note will cover the most popular temperature sensor technologies to a level of detail that will give the reader insight into how to determine which sensor is most appropriate for the application. This note is written from the perspective of catering to the complex issues of the sensing environment and required accuracy. Once the sensor is selected, subsequent Microchip application notes can be used to design appropriate microcontroller interface circuits. These circuits will offer the complete signal path from the low level output signals of the sensor, through the analog signal conditioning stages to the microcontroller. Techniques such as sensor excitation, sensor signal gain, and digital linearization are reserved for these further discussions.

SO MANY TEMPERATURE SENSORS

The most popular temperature sensors used today are the Thermocouple, Resistive Temperature Device (RTD), Thermistor, and the newest technology, the Integrated Silicon Based Sensors. There are other sensing technologies, such as Infrared (Pyrometers) and Thermal Pile. These alternatives are beyond the scope of this application note.

Each of these sensor technologies cater to specific temperature ranges and environmental conditions. The sensor's temperature range, ruggedness, and sensitivity are just a few characteristics that are used to determine whether or not the device will satisfy the requirements of the application. No one temperature sensor is right for all applications. The thermocouple's wide temperature range is unrivaled as is the excellent linearity of the RTD and the accuracy of the Thermistor.

Table 1 summarizes the main characteristics of these four temperature sensors. This table can be used during the first pass of the sensor selection process. Further details concerning the construction and characteristics of these sensors are given in the following sections of this application note.

To complement the specifications cited in Table 1, a list of typical applications for these four temperature sensors are shown in Table 2.

	Thermocouple	RTD	Thermistor	Integrated Silicon
Temperature Range	-270 to 1800°C	-250 to 900 °C	-100 to 450°C	-55 to 150°C
Sensitivity	10s of $\mu\text{V} / ^\circ\text{C}$	0.00385 $\Omega / \Omega / ^\circ\text{C}$ (Platinum)	several $\Omega / \Omega / ^\circ\text{C}$	Based on technology that is -2mV/°C sensitive
Accuracy	$\pm 0.5^\circ\text{C}$	$\pm 0.01^\circ\text{C}$	$\pm 0.1^\circ\text{C}$	$\pm 1^\circ\text{C}$
Linearity	Requires at least a 4th order polynomial or equivalent look up table.	Requires at least a 2nd order polynomial or equivalent look up table.	Requires at least 3rd order polynomial or equivalent look up table.	At best within $\pm 1^\circ\text{C}$. No linearization required.
Ruggedness	The larger gage wires of the thermocouple make this sensor more rugged. Additionally, the insulation materials that are used enhance the thermocouple's sturdiness.	RTDs are susceptible to damage as a result of vibration. This is due to the fact that they typically have 26 to 30 AWG leads which are prone to breakage.	The thermistor element is housed in a variety of ways, however, the most stable, hermetic Thermistors are enclosed in glass. Generally thermistors are more difficult to handle, but not affected by shock or vibration.	As rugged as any IC housed in a plastic package such as dual-in-line or surface outline ICs.
Responsiveness in stirred oil	less than 1 Sec	1 to 10 Secs	1 to 5 Secs	4 to 60 Secs
Excitation	None Required	Current Source	Voltage Source	Typically Supply Voltage
Form of Output	Voltage	Resistance	Resistance	Voltage, Current, or Digital
Typical Size	Bead diameter = 5 x wire diameter	0.25 x 0.25 in.	0.1 x 0.1 in.	From TO-18 Transistors to Plastic DIP
Price	\$1 to \$50	\$25 to \$1000	\$2 to \$10	\$1 to \$10

TABLE 1: The most common temperature sensors in industry are the thermocouple, RTD, thermistor, and integrated silicon based. No one temperature sensor is right for all applications. The thermocouple's wide temperature range is unrivalled as is the excellent linearity of the RTD and the accuracy of the thermistor. The silicon sensor is easy to implement and install in a circuit.

Sensor Type	Application
Thermocouple	Extremely high temperature sensing, biophysics, metal cutting research, gas chromatography, internal combustion engine temperatures, chemical reactions
RTD	Cold junction compensation, bridge temperature, calibration, process control.
Thermistor	Cold junction compensation, bridge temperature sensing, pyrometer calibration, vacuum manometers, anemometers, flow meters, liquid level, fluid velocity, thermal conductivity cells, gas chromatography
Silicon Based	Cold junction compensation, personal computers, office electronics, cellular phones, HVAC, battery management, four speed controls

TABLE 2: Listed are some examples of the applications that each temperature sensor is best suited for.

THE VERSATILE, INEXPENSIVE THERMOCOUPLE

The thermocouple consists of two wires of dissimilar metals that are soldered together at one end as shown in Figure 1. The temperature at the Reference Junction (also known as the Cold Junction Compensation Point) is used to negate the errors contributed by the Iron-Copper and Constantan-Copper junctions. The connecting point of the two metals of the thermocouple is positioned on the target where the temperature measurement is needed.

This configuration of materials produces a voltage between the two wires at the unsoldered end that is a function of the temperature of all of the junctions. Consequently, the thermocouple does not require voltage or current excitation. As a matter of fact, an attempt to provide either type of excitation could introduce errors into the system.

Since a voltage develops at the open end of the two dissimilar wires, it would seem as if the thermocouple interface could be done in a straight forward manner by measuring the voltage difference between the wires.

This could easily be the case if it wasn't for the fact that the termination ends of the thermocouple wires connect to another metal, usually copper.

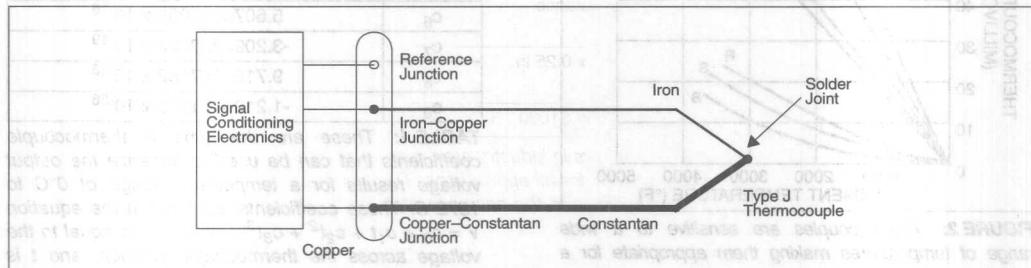


FIGURE 1: A thermocouple is constructed of two dissimilar metals, such as the Iron and Constantan in this Type J thermocouple. The temperature of the Reference Junction Compensation (also known as the Cold Junction Compensation or Isothermal Block) is used to negate the errors contributed by the Iron-Copper and Constantan-Copper Junctions.

This creates another pair of thermocouples, which introduces a significant error to the system. The only way to negate this error is to sense the temperature at the Reference Junction box (Figure 1) and subtract the contributing errors of these connections in a hardware solution or a combination of software and hardware.

Pure hardware calibration techniques are more limited in terms of linearization correction than the combination of software and hardware techniques. Typically, an RTD, Thermistor, or Integrated Silicon Sensor is used to sense this junction temperature accurately.

In principle the thermocouple can be made from any two metals, however, in practice standard combinations of these two metals have been embraced because of their desirable qualities of linearity and their voltage magnitude drop versus temperature. These common thermocouple types are E, J, T, K, N, S, B, and R (summarized in Table 3 and Figure 2).

Thermocouples are highly non-linear and require significant linearization algorithms, as will be discussed later. The Seebeck Coefficient in Table 3 represents the average drift of the specific thermocouple at a specific temperature.

Thermocouple Type	Conductors	Temperature Range (°C)	Seebeck Coefficient	Application Environments
E	Chromel, Constantan	-200 to 900	60µV/°C	oxidizing, inert, vacuum
J	Iron, Constantan	0 to 760	51µV/°C	vacuum, oxidizing reducing, inert
T	Copper, Constantan	-200 to 371	40µV/°C	corrosive, moist, subzero
K	Chromel, Alumel	-200 to 1260	40µV/°C	completely inert
N	Nicrosil, Nisil	0 to 1260	38µV/°C	oxidizing
S	Platinum(10% Rhodium), Platinum	0 to 1480	11µV/°C	oxidizing, inert
B	Platinum (30% Rhodium) Platinum (6% Rhodium)	0 to 1820	8µV/°C	oxidizing, inert
R	Platinum (13% Rhodium), Platinum	0 to 1480	12µV/°C	oxidizing, inert

TABLE 3: The most common thermocouple types are shown with their standardized material and performance specifications. These thermocouple types are fully characterized by the American Society for Testing and Materials (ASTM) and specified in IST-90 units per NIST Monograph 175.

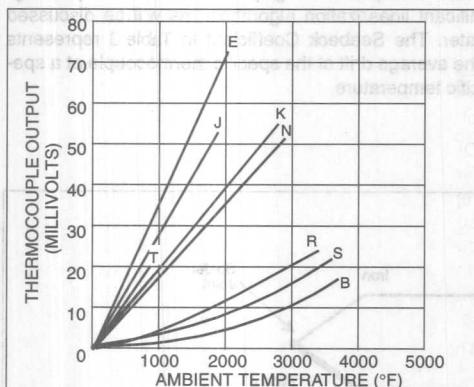


FIGURE 2: Thermocouples are sensitive to a wide range of temperatures making them appropriate for a variety of hostile environments.

At the time of shipment, the thermocouple performance is guaranteed by the vendor in accordance with NIST 175 standards (adopted by ASTM). These standards define the temperature behavior of the thermocouple as well as the quality of the material used.

Thermocouples are extremely non-linear when compared to RTD, Thermistor, and Integrated Silicon Sensors. Consequently, complex algorithms must be performed with the processor portion of the circuit. An example of the complexity of the calculation is shown in Table 4. These are the Type K Thermocouple coefficients that can be used to linearize the output voltage results for a temperature range of 0°C to 1372°C. These coefficients are used in the equation

$$V = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \dots$$

where

V is equal to the voltage across the thermocouple junction, and

t is equal to the temperature.

c ₀	-1.7600413686 x 10 ⁻²
c ₁	3.8921204975 x 10 ⁻²
c ₂	1.8558770032 x 10 ⁻⁵
c ₃	-9.9457592874 x 10 ⁻⁸
c ₄	3.1840945719 x 10 ⁻¹⁰
c ₅	-5.6072844889 x 10 ⁻¹³
c ₆	5.6075059059 x 10 ⁻¹⁶
c ₇	-3.2020720003 x 10 ⁻¹⁹
c ₈	9.7151147152 x 10 ⁻²³
c ₉	-1.2104721275 x 10 ⁻²⁶

TABLE 4: These are the Type K thermocouple coefficients that can be used to linearize the output voltage results for a temperature range of 0°C to 1372°C. These coefficients are used in the equation $V = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \dots$ where V is equal to the voltage across the thermocouple junction, and t is equal to the temperature.

The alternative to using these complex calculations is to use program memory for a look-up table. The replacement look-up table for the equation coefficients of the Type K thermocouple in Table 4 is approximately an 11 x 14 array of decimal integers ranging from 0.000 to 13.820.

Additionally, the thermocouple can quantify temperature as it relates to a reference temperature. The reference temperature is defined as the temperature at the end of the thermocouple wires furthest from the soldered bead. This reference temperature is usually sensed using an RTD, Thermistor, or Integrated Silicon Sensor.

The thermal mass of the thermocouple is smaller than the RTD or Thermistor, consequently the response of the thermocouple as compared to larger temperature sensors is faster. The wide temperature ranges of the sensor makes it exclusively appropriate for many hostile sensing environments.

Thermocouple Error Analysis

Thermocouples are generally low cost, rugged and available in smaller sizes than the other temperature sensors. Any stress on the material due to bending stretching or compression can change the characteristics of the thermal gradients. Additionally, corrosive material can penetrate the insulation material and cause a change in the thermal characteristics. It is possible to encase the thermocouple bead in protective tubing such as a ceramic tube for high temperature protection. Metallic wells can also provide mechanical protection.

The thermocouple voltage drop occurs along the temperature gradient down the length of the two dissimilar metals. This does not imply that shorter versus longer wires will necessarily have differing Seebeck Coefficients. With shorter wires, the temperature gradient is simply steeper. However, the longer wires do have an advantage in terms of conduction affects. With the longer wires the temperature gradient is lower and conduction losses are reduced.

On the down side, these types of temperature sensors have a very low output signal. This places additional requirements on the signal conditioning circuitry that follows the thermocouple. In addition to this low level output signal, the linearity of the device requires a considerable amount of calibration. This calibration is typically done in firmware as well as software. In firmware, an absolute temperature reference is needed which serves as a "cold junction" reference. In software, the linearity errors of the thermocouple are reduced with look-up tables or high order polynomial equations. And finally, EMI signals are easily coupled in to this two-wire system.

Lower gage wires are required for higher temperatures and will also have a longer life. However, if sensitivity is a prime concern, larger wire gages will provide better measurement results.

To summarize, thermocouples are usually selected because for the wide temperature range, ruggedness, and price. Accuracy and good linearity are hard to achieve in precision systems. If high accuracy is desirable, other temperature sensors may be a better alternative.

THE RTD IS ABSOLUTELY AN ALTERNATIVE

RTD element technologies are constantly improving, enhancing the quality of the temperature measurement. To produce a high quality, accurate temperature measurement system, the selection of the RTD element is critical. The RTD (Resistance Temperature Detector) is a resistive element constructed from metals, such as, Platinum, Nickel or Copper. The particular metals that are chosen exhibit a predictable change in resistance with temperature. Additionally, they have the basic physical properties that allow for easy fabrication. The temperature coefficient of resistance of these metals is large enough to render measurable changes with temperature.

Other temperature sensing devices, such as thermocouples, fall short of giving the designer an absolute result that is fairly linear over temperature. The linear relation between resistance and temperature of the RTD simplifies the implementation of signal conditioning circuitry. The resistance change to temperature of each of these types of RTDs is shown in Table 5. Platinum RTDs (PRTD) are the most accurate and reliable of the three types shown in Table 5.

Of all the material types, Platinum RTDs are best suited for precision applications where absolute accuracy and repeatability is critical. The platinum material is less susceptible to environmental contamination, where copper is prone to corrosion causing long term stability problems. Nickel RTDs tolerate environmental conditions fairly well, however, they are limited to smaller temperature ranges.

The PRTD has nearly linear thermal response, good chemical inertness and is easy to manufacture in the form of small-diameter wires or films. As shown in Table 5, the resistivity of the platinum is higher than the other metals, making the physical size of the element smaller. This offers advantages where "real-estate" is at a premium as well better thermal responsiveness.

Thermal responsiveness of an RTD affects the measurement time. It is also dependent on the housing material of the RTD and the size of the implementation of the RTD element. Elements with smaller dimensions can be housed in smaller packages. Since RTD are typically smaller, their thermal response times can be shorter than silicon based temperature sensors.

The absolute, 0°C value of the element is available in a wide range of resistances and can be specified by the user. For instance, the standard resistance of a platinum RTD (PRTD) is 100Ω. But, they are also available as 50, 100, 200, 500, 1000 or 2000Ω elements.

As stated before, the RTD is an absolute temperature sensing devices as opposed to the thermocouple, which senses relative temperatures. Consequently, additional temperature sensors would not necessarily enhance the accuracy of the system.

RTD Detector Material	Thermal Response (at 0°C)	Typical Material Resistivity (at 0°C)
Platinum	0.00385 Ω/Ω°C (IEC 751)	9.81 × 10 ⁻⁶ Ω cm
Nickel	0.00672 Ω/Ω°C	5.91 × 10 ⁻⁶ Ω cm
Copper	0.00427 Ω/Ω°C	1.53 × 10 ⁻⁶ Ω cm

TABLE 5: RTD temperature sensing devices are available in a variety of materials. The temperature coefficient of these devices is specified in terms of ohms, per ohms per °C.

In most applications, linearization is not required. Table 6 shows the temperature versus resistance of a 100 Ω platinum RTD. With a 100Ω PRTD, the change in resistance from 0°C to 100°C changes resistance by:

$$\Delta R = (\text{Thermal Response}) \times R_0 \times \Delta t$$

$$\Delta R = 0.00038 \Omega/\Omega°C \times 100\Omega \times 100°C$$

$$\Delta R = 38.5\Omega$$

The accuracy of the PRTD over its temperature range is also shown in terms of Δ°C from ideal.

Of the temperature sensors discussed in this application note, the RTD is the most linear with only two coefficients in the linearization equation,

$$R_t = R_0(I + At + Bt^2)$$

for temperatures 0 °C to 859 °C

$$R_t = R_0(I + At + Bt^2) + C(t - 100t^3)$$

for temperatures -200°C to 0°C

where

R_t is the resistance of the RTD at measurement

temperature,

t is the temperature being measured,

R₀ is the magnitude of the RTD at 0°C,

A, B and C are calibration coefficients derived from experimentation.

These equations are solved after five iterations making it possible to resolve to ±0.001°C of accuracy.

RTD Error Analysis

Beyond the initial element errors shown in Table 6 there are other sources of error that effect the overall accuracy of the temperature sensor. The introduction of defects into the mechanical integrity of the part such as bending the wires, shock due to rough handling, constriction of the packaging that leads to stress during thermal expansion, and vibration can have a long term effect on the repeatability of the sensor.

Although the mechanical stresses can effect long term stability, the electrical design used to condition, gain and digitize the RTD output can also effect the overall accuracy. One of these sources of errors is the self heating of the RTD element that results from the required current excitation. A current excitation is used to convert the resistance of the RTD into a voltage. It is desirable to have a high excitation current through the resistive sensing element in order to keep the output voltage above the system noise levels. A negative side to this design approach is that the element will self-heat as a result of the higher current. The combination of current and resistance create power and in turn the by-product of heat. The heat generated by the power dissipation of the element artificially increases the resistance of the RTD.

The error contribution of the heat generated by the element's power dissipation is easily calculated given the package thermal resistance ($\theta_{PACKAGE}$), the magnitude of the current excitation and the value of the RTD resistance (R_{RTD}).

Temperature (°C)	Typical Absolute Resistive Value (Ω)	Deviation in Ω	Deviation in °C
-200	23.0	± 0.56	± 1.3
-100	61.5	± 0.32	± 0.8
0	100.0	± 0.12	± 0.3
100	138.5	± 0.30	± 0.8
200	177.0	± 0.48	± 1.3
300	215.5	± 0.64	± 1.8
400	254.0	± 0.79	± 2.3
500	292.5	± 0.93	± 2.8
600	331.0	± 1.06	± 3.3
700	369.5	± 1.17	± 3.8
800	408.0	± 1.28	± 4.3

TABLE 6: OMEGA Platinum Resistance Elements Allowable Deviation from Ideal Values for a 100Ω Sensor. The PRTD in this illustration is manufactured to have a thermal response of 0.00385Ω/Ω / °C (IEC 751) near 0°C, Class B.

For example, if the package thermal resistance is $50^{\circ}\text{C}/\text{W}$, the RTD's nominal resistance is 250Ω , and the element is excited with a 5mA current source, the artificial increase in temperature ($\Delta^{\circ}\text{C}$) as a result of self heating is:

$$\begin{aligned}\Delta^{\circ}\text{C} &= I^2 R_{RTD} * \theta_{PACKAGE} \\ \Delta^{\circ}\text{C} &= (5\text{mA})^2 \times 250\Omega \times 50^{\circ}\text{C}/\text{Watt} \\ \Delta^{\circ}\text{C} &= 0.3125^{\circ}\text{C}\end{aligned}$$

This example illustrates the importance of keeping the magnitude of current excitation as low as possible, preferably less than 1mA .

A second source of error resulting from the electrical design comes from the lead wires to and from the sensing element. The technique used to connect the RTD to the rest of the circuit can be a critical issue. Three possible wire configurations can be used when connecting the element to the remainder of the circuit. In Figure 3a, the 2-wire configuration is by far the least expensive, however, the current that is used to excite the RTD element flows through the wires as well are the resistive element. A portion of the wires are exposed to the same temperatures as the RTD. The effects of the wire resistance change with temperature can become a critical issue.

For example, if the lead wire is constructed of 5 gage copper leads that are 50 meters long (with a wire resistance of $1.028\Omega/\text{km}$), the contribution of both wires increases the RTD resistance by 0.1028Ω . This translates into a temperature measurement error of 0.26°C for a 100Ω @ 0°C RTD. This error contributes to the non-linearity of the overall measurement. The least accurate of configurations shown in Figure 3 is the 2-wire. Circuits can be configured to effectively use the 3-wire and 4-wire configuration to remove the error contribution of the lead wires completely.

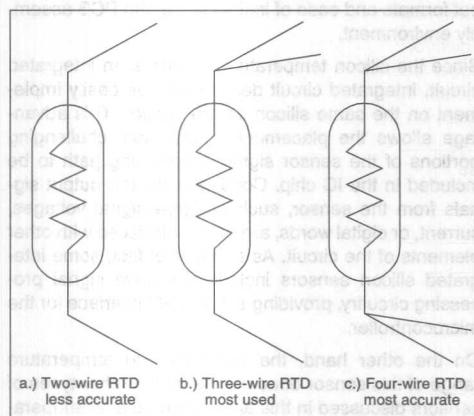


FIGURE 3: RTD elements are available in two-wire, three-wire or four-wire configurations. Two-wire RTDs are the least accurate because the contribution of the wire resistance and wire resistance drift to the measurement. With four-wire RTDs, this error can be eliminated by using force and sense techniques in the circuit design.

GET THE GREAT ACCURACY OF THE THERMISTOR

If accuracy is a high priority, the thermistor should be the temperature sensor of choice. Thermistors are available in two varieties, NTC and PTC. The NTC (negative temperature coefficient) thermistor is constructed of ceramics composed of oxides of transition metals (manganese, cobalt, copper, and nickel). With a current excitation the NTC has a negative temperature coefficient that is very repeatable and fairly linear. These temperature dependent semiconductor resistors operate over a range for -100°C to 450°C . Combined with the proper packaging, they have a continuous change of resistance over temperature. This resistive change versus temperature is larger than the RTD (see Figure 4), consequently the thermistor is systematically more sensitive.

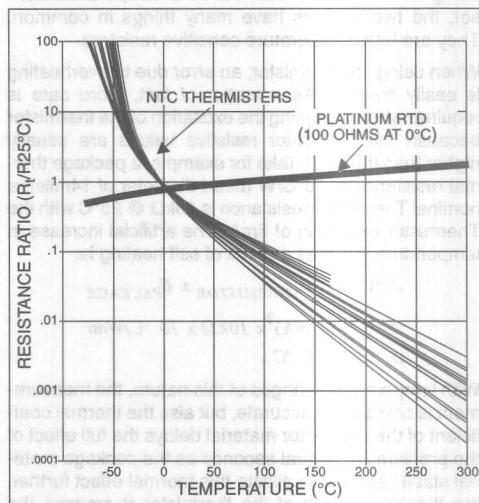


FIGURE 4: The temperature response versus resistance of the NTC thermistor and the RTD.

The temperature characteristics of a typical NTC thermistor along with a 100Ω RTD is shown in Figure 4. In this figure, the difference between the temperature coefficients of these two sensors is noticeable. The thermistor has a negative temperature coefficient as expected and the absolute value of the sensor changes by 10,000 times over its usable temperature range. In contrast, the RTD shown has a positive temperature coefficient and only changes by four times over its usable temperature range. This higher sensitivity of the thermistor makes it attractive in terms of accuracy in measurements.

The Thermistor is less linear than the RTD in that it requires a 3rd order polynomial for precise temperature corrections. The linearity equations for the Thermistor are:

$$\ln R_T = B_0 + \frac{B_1}{t} + \frac{B_2}{t^2} + \frac{B_3}{t^3}$$

over the entire temperature range
where

B_X are the material constants of the thermistor

This linearization formula can resolve to a total measurement uncertainty of $\pm 0.005^\circ\text{C}$. However, it is tedious when implemented in the microcontroller. Alternatively, look-up tables can be generated to serve the same purpose with slightly less accuracy.

Thermistor Error Analysis

Although the NTC thermistor has the capability of being more accurate than the RTD temperature sensor, the two sensors have many things in common. They are both temperature sensitive resistors.

When using the thermistor, an error due to overheating is easily created. As a matter of fact, more care is required when designing the excitation of the thermistor because the thermistor resistive values are usually higher than the RTD. Take for example, a package thermal resistance of $10^\circ\text{C}/\text{W}$ (bead diameter of 14mils), a nominal Thermistor resistance is $10\text{k}\Omega$ @ 25°C with the Thermistor excitation of 5mA. The artificial increase in temperature (ΔT) as a result of self heating is:

$$\Delta T = I^2 R_{THERMISTOR} \times \theta_{PACKAGE}$$

$$\Delta T = (5\text{mA})^2 \times 10\text{k}\Omega \times 10^\circ\text{C}/\text{Watt}$$

$$\Delta T = 2.5^\circ\text{C}$$

With temperature changes of this nature, the measurement is obviously inaccurate, but also the thermal coefficient of the thermistor material delays the full effect of the problem for several seconds as the package material stabilizes. To complicate this thermal effect further, the thermal heating of the thermistor decreases the thermistor resistance (instead of the increase seen with the RTD). Since the thermistor has a negative resistive coefficient, the overheating effect reverses as the thermistor resistance becomes less than the voltage across the thermistor divided by the excitation current. This phenomena is not easily overcome with software calibration and should be avoided.

The PTC thermistor has a positive temperature coefficient and is constructed from barium titanate. The sensitivity of the PTC is considerably higher than the sensitivity of the NTC thermistor and should be used when a specific temperature range is of interest (-25 to 150°C). Over the lower portion of the resistance versus temperature curve the thermistor resistance is fairly constant. At higher temperatures the material passes through a threshold temperature (between 80°C and 140°C , dependent on chemical composition of the ceramic) where the resistance versus temperature characteristics change dramatically (Figure 5).

At this point, increases in temperature cause a rise in the PTC's resistance and the PTC resistive / temperature characteristics become very steep.

A second type of PTC thermistor is known as the Silistor. This device is constructed of a thermally sensitive silicon material and also has a positive temperature coefficient (-60°C to 150°C) that is linear over the entire operating range.

Both of the thermal characteristics of the PTC type thermistors are shown in Figure 5.

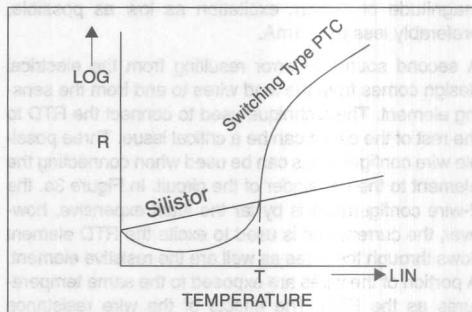


FIGURE 5: PTC thermistor and silistor resistance versus temperature response.

SELECT THE EASY TO USE INTEGRATED SILICON TEMPERATURE SENSOR

The integrated circuit temperature sensors offer another alternative to solving temperature measurement problems. The advantages of integrated circuit silicon temperature sensors include, user friendly output formats and ease of installation in the PCB assembly environment.

Since the silicon temperature sensor is an integrated circuit, integrated circuit designs can be easily implemented on the same silicon as the sensor. This advantage allows the placement of the most challenging portions of the sensor signal conditioning path to be included in the IC chip. Consequently, the output signals from the sensor, such as large signal voltages, current, or digital words, are easily interfaced with other elements of the circuit. As a matter of fact, some integrated silicon sensors include extensive signal processing circuitry, providing a digital I/O interface for the microcontroller.

On the other hand, the accuracy and temperature range of this sensor does not match the other types of sensors discussed in this application note. A temperature sensor IC can operate over a nominal temperature range of -55 to 150°C . Some devices go beyond this range, while others operate over a narrower range.

CHOOSE THE RIGHT TEMPERATURE SENSOR

Of the temperature sensors on the market today, the thermocouple, RTD, Thermistor, and Integrated Silicon Sensors are continuing to dominate. The thermocouple is most appropriate for higher temperature sensing, while the RTD is best suited for lower temperatures were good linearity is desirable. The Thermistor is typically used for applications with smaller temperature ranges, but it offers greater accuracy than the thermocouple or the RTD.

All four of the sensors mentioned in this application note have the capability of providing good, accurate, and reliable performance, making the final sensor selection appear somewhat trivial. However, once the temperature sensor has been selected, the next step is to design the analog and digital signal conditioning circuit. The design of this circuit will determine the actual performance that is finally achieved.

Several application notes can be found in the Microchip's library that elaborate on these circuits. Each of these application notes will present circuit alternatives that take into account simplicity, accuracy and cost.

REFERENCES

- Baker, Bonnie, "Low Power Temperature Sensing with Precision Converters", *Sensors*, (February 1997) p 38.
- Baker, Bonnie, "Precision Temperature Sensing with RTD Circuits", AN687, Microchip Technology Inc. (1998).
- Baker, Bonnie, "Single Supply Temperature Sensing with Thermocouples", AN684, Microchip Technology Inc. (1998).
- Baker, Bonnie, "Thermistors in Single Supply Temperature Sensing Circuits", AN685, Microchip Technology Inc. (1998).
- Klopfenstein, Rex, "Software Linearization of a Thermocouple", *Sensors*, (December 1997) p 40.
- Product Book, *Thermometrics, Inc.* (1997).
- Schraff, Fred. "Thermocouple Basics" *Measurement & Control*, (June 1996) p 126.
- Sulciner, James, "Understanding and Using PRTD Technology, Part 1: History, Principles and Designs", *Sensors*, (August 1996).
- <http://www.omega.com/techref/>

NOTES:



MICROCHIP

AN682

Using Single Supply Operational Amplifiers in Embedded Systems

Author: Bonnie Baker
Microchip Technology Inc.

INTRODUCTION

Beyond the primitive transistor, the operational amplifier is the most basic building block for analog applications. Fundamental functions such as gain, load isolation, signal inversion, level shifting, adding and/or subtracting signals are easily implemented with this building block. More complex circuits can also be implemented, such as the instrumentation amplifier, a current to voltage converter, and filters, to name only a few. Regardless of the level of complexity of the operational amplifier circuit, knowing the fundamental operation and behavior of this building block will save a considerable amount of upfront design time.

Formal classes on this subject can be very comprehensive and useful. However, many times they fall short in terms of experience or common sense. For instance, a common mistake that is made when designing with operational amplifiers is to neglect to include the bypass capacitors in the circuit. Operational amplifier theory often overlooks this practical detail. If the bypass capacitor is missing, the amplifier circuit could oscillate at a frequency that "theoretically" doesn't make sense. If text book solutions are used, this is a difficult problem to solve.

This application note is divided into three sections. The first section will list the fundamental amplifier applications with the design equations included. These amplifier circuits were selected with embedded system integration in mind.

The second section will use these fundamental circuits to build useful amplifier functions in embedded control applications.

The third section will identify the most common single supply operational amplifier (op amp) circuit design mistakes. This list of mistakes have been gathered over many years of trouble shooting circuits with numerous designers in the industry. The most common design pitfalls can easily be avoided if the check list from this short tutorial is used.

FUNDAMENTAL OPERATIONAL AMPLIFIER CIRCUITS

The op amp is the analog building block that is analogous to the digital gate. By using the op amp in the design, circuits can be configured to modify the signal in the same fundamental way that the inverter, AND, and OR gates do in digital circuits. In this section, fundamental building blocks such as the voltage follower, non-inverting gain and inverting gain circuits will be discussed. This will be followed by a rail splitter, difference amplifier, summing amplifier and current to voltage converter.

Voltage Follower Amplifier

Starting with the most basic op amp circuit, the buffer amplifier (shown in Figure 1) is used to drive heavy loads, solve impedance matching problems, or isolate high power circuits from sensitive, precise circuitry.

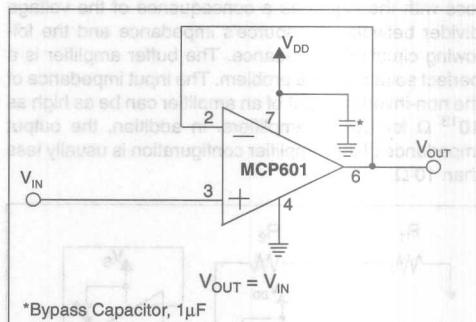


FIGURE 1: Buffer Amplifier; also called a voltage follower.

The buffer amplifier, shown in Figure 1, can be implemented with any single supply, unity gain stable amplifier. In this circuit as with all amplifier circuits, the op amp must be bypassed with a capacitor. For single supply amplifiers that operate in bandwidths from DC to megahertz, a $1\mu\text{F}$ capacitor is usually appropriate. Sometimes a smaller bypass capacitor is required for amplifiers that have bandwidths up to the 10s of megahertz. In these cases a $0.1\mu\text{F}$ capacitor would be appropriate. If the op amp does not have a bypass capacitor or the wrong value is selected, it may oscillate.

AN1002

feedback loop is tied from the output of the amplifier to the inverting input. An all too common error is to assume that an op amp circuit that has a positive gain requires positive feedback. If positive feedback is used, the amplifier will most likely drive to either rail at the output.

This amplifier circuit will give good linear performance across the bandwidth of the amplifier. The only restrictions on the signal will occur as a result of a violation of the input common-mode and output swing limits. These limitations will be discussed in the third section of this application note ("Amplifier Design Pitfalls").

If this circuit is used to drive heavy loads, the amplifier that is actually selected must be specified to provide the required output currents. Another application where this circuit may be used is to drive capacitive loads. Not every amplifier is capable of driving capacitors without becoming unstable. If an amplifier can drive capacitive loads, the product data sheet will highlight this feature. However, if an amplifier can't drive capacitive loads, the product data sheets will not explicitly say.

Another use for the buffer amplifier is to solve impedance matching problems. This would be applicable in a circuit where the analog signal source has a relatively high impedance as compared to the impedance of the following circuitry. If this occurs, there will be a voltage loss with the signal as a consequence of the voltage divider between the source's impedance and the following circuitry's impedance. The buffer amplifier is a perfect solution to the problem. The input impedance of the non-inverting input of an amplifier can be as high as $10^{13} \Omega$ for CMOS amplifiers. In addition, the output impedance of this amplifier configuration is usually less than 10Ω .

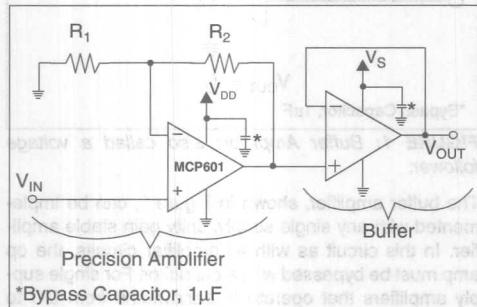


FIGURE 2: Load isolation is achieved using a buffer amplifier.

Yet another use of this configuration is to separate a heat source from sensitive precision circuitry, as shown in Figure 2. Imagine that the input circuitry to this buffer amplifier is amplifying a $100\mu\text{V}$ signal. This type of amplification is difficult to do with any level of accuracy in the best of situations. This precision measurement can easily be disrupted by changing the output current drive of the device that is doing the amplification work.

the chip which will induce an offset change. An analog buffer can be used to perform the function of driving heavy loads while the front end circuitry can be used to make precision measurements.

Gaining Analog Signals

The buffer solves a lot of analog signal problems, however, there are instances in circuits where a signal needs to be gained. Two fundamental types of amplifier circuits can be used. With the first type, the signal is not inverted as shown in Figure 3. This type of circuit is useful in single supply¹ amplifier applications where negative voltages are usually not possible.

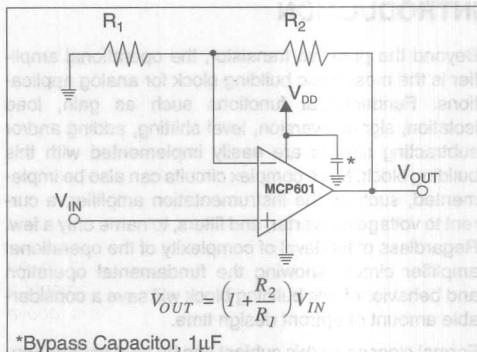


FIGURE 3: Operational amplifier configured in a non-inverting gain circuit.

The input signal to this circuit is presented to the high impedance, non-inverting input of the op amp. The gain that the amplifier circuit applies to the signal is equal to:

$$V_{OUT} = \left(1 + \frac{R_2}{R_1}\right) V_{IN}$$

Typical values for these resistors in single supply circuits are above $2k\Omega$ for R_2 . The resistor, R_1 , restrictions are dependent on the amount of gain desired versus the amount of amplifier noise and input offset voltage as specified in the product data sheet of the op amp.

Once again, this circuit has some restrictions in terms of the input and output range. The non-inverting input is restricted by the common-mode range of the amplifier. The output swing of the amplifier is also restricted as stated in the product data sheet of the individual amplifier. Most typically, the larger signal at the output of the amplifier causes more signal clipping errors than the smaller signal at the input. If undesirable clipping occurs at the output of the amplifier, the gain should be reduced.

1. For this discussion, single supply implies that the negative supply pin of the operational amplifier is tied to ground and the positive supply pin is tied to $+5\text{V}$. All discussion in this application note can be extrapolated to other supply voltages where the single supply exceeds 5V or dual supplies are used.

An inverting amplifier configuration is shown in Figure 4. With this circuit, the signal at the input resistor, R_1 , is gained and inverted to the output of the amplifier. The gain equation for this circuit is:

$$V_{OUT} = -\left(\frac{R_2}{R_1}\right)V_{IN} + \left(1 + \frac{R_2}{R_1}\right)V_{BIAS}$$

The ranges for R_1 and R_2 are the same as in the non-inverting circuit shown in Figure 3.

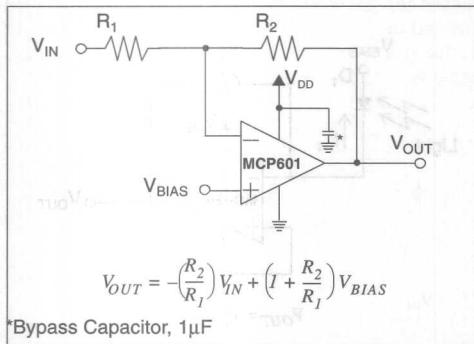


FIGURE 4: Operational amplifier configured in an inverting gain circuit. In single supply environments a V_{BIAS} is required to insure the output stays above ground.

In single supply applications, this circuit can easily be misused. For example, let R_2 equal 10kΩ, R_1 equal 1kΩ, V_{BIAS} equal 0V, and the voltage at the input resistor, R_1 , equal to 100mV. With this configuration, the output voltage would be -1V. This would violate the output swing range of the operational amplifier. In reality, the output of the amplifier would go as near to ground as possible.

The inclusion of a DC voltage at V_{BIAS} in this circuit solves this problem. In the previous example, a voltage of 225mV applied to V_{BIAS} would level shift the output signal up 2.475V. This would make the output signal equal (2.475V - 1V) or 1.475V at the output of the amplifier. Typically, the average output voltage should be designed to be equal to $V_{DD}/2$.

Single Supply Circuits and Supply Splitters

As was shown in the inverting gain circuit (Figure 4), single supply circuits often need a level shift to keep the signal between negative (usually ground) and positive supply pins. This level shift can be designed with a single amplifier and a combination of resistors and capacitors as shown in Figure 5. Many times a simple buffer amplifier without compensation capacitors will accomplish this task. In other cases the level shift circuit will see dynamic or transient load changes, like the reference to an Analog-to-Digital (A/D) converter. In these applications, the level shift circuit must hold its voltage constant. If it does change, a conversion error might be observed.

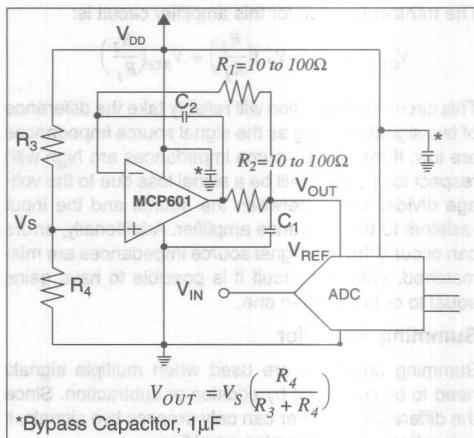


FIGURE 5: A supply splitter is constructed using one operational amplifier. This type of function is particularly useful in single supply circuits.

A solid level shift voltage can easily be implemented using a voltage divider (R_3 and R_4) or a reference voltage source buffered by the amplifier. The transfer function for this circuit is:

$$V_{OUT} = V_S \left(\frac{R_4}{R_3 + R_4} \right)$$

The circuit in Figure 5 has an elaborate compensation scheme to allow for the heavy capacitive load, C_1 . The benefit of this big capacitor is that it presents a very low AC resistance to the reference pin of the A/D converter. In the AC domain, the capacitor serves as a charge reservoir that absorbs any momentary current surges which are characteristic of sampling A/D converter reference pins.

The Difference Amplifier

The difference amplifier combines the non-inverting amplifier and inverting amplifier circuits of Figure 3 and Figure 4 into a signal block that subtracts two signals. The implementation of this circuit is shown in Figure 6.

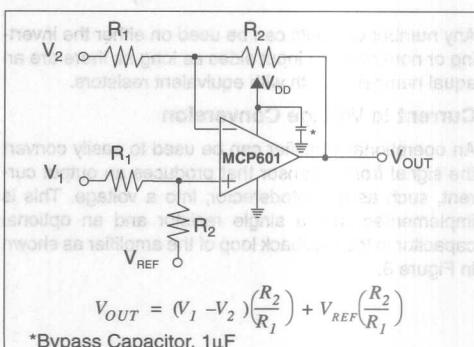


FIGURE 6: Operational amplifier configured in a difference amplifier circuit.

The transfer function for this amplifier circuit is:

$$V_{OUT} = (V_1 - V_2) \left(\frac{R_2}{R_1} \right) + V_{REF} \left(\frac{R_2}{R_1} \right)$$

This circuit configuration will reliably take the difference of two signals as long as the signal source impedances are low. If the signal source impedances are high with respect to R_1 , there will be a signal loss due to the voltage divider action between the source and the input resistors to the difference amplifier. Additionally, errors can occur if the two signal source impedances are mismatched. With this circuit it is possible to have gains equal to or higher than one.

Summing Amplifier

Summing amplifiers are used when multiple signals need to be combined by addition or subtraction. Since the difference amplifier can only process two signals, it is a subset of the summing amplifier.

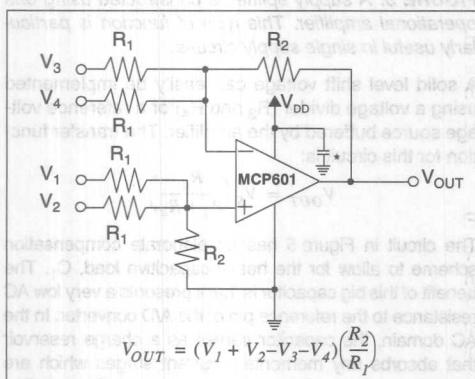


FIGURE 7: Operational amplifier configured in a summing amplifier circuit.

The transfer function for this circuit is:

$$V_{OUT} = (V_1 + V_2 - V_3 - V_4) \left(\frac{R_2}{R_1} \right)$$

Any number of inputs can be used on either the inverting or non-inverting input sides as long as there are an equal number of both with equivalent resistors.

Current to Voltage Conversion

An operational amplifier can be used to easily convert the signal from a sensor that produces an output current, such as a photodetector, into a voltage. This is implemented with a single resistor and an optional capacitor in the feedback loop of the amplifier as shown in Figure 8.

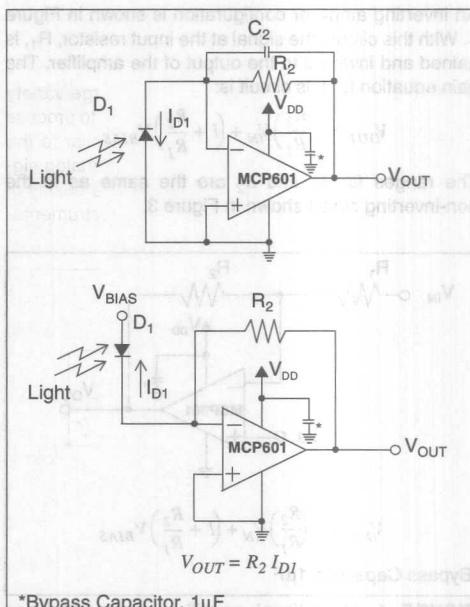


FIGURE 8: Current to voltage converter using an amplifier and one resistor. The top light scanning circuit is appropriate for precision applications. The bottom circuit is appropriate for high speed applications.

As light impinges on the photo diode, charge is generated, causing a current to flow in the reverse bias direction of the photodetector. If a CMOS op amp is used, the high input impedance of the op amp causes the current from the detector (ID1) to go through the path of lower resistance, R_2 . Additionally, the op amp input bias current error is low because it is CMOS (typically <200 pA). The non-inverting input of the op amp is referenced to ground which keeps the entire circuit biased to ground. These two circuits will only work if the common mode range of the amplifier includes zero.

Two circuits are shown in Figure 8. The top circuit is designed to provide precision sensing from the photodetector. In this circuit the voltage across the detector is nearly zero and equal to the offset voltage of the amplifier. With this configuration, current that appears across the resistor, R_2 , is primarily a result of the light excitation on the photodetector.

The photosensing circuit on the bottom of Figure 8 is designed for higher speed sensing. This is done by reverse biasing the photodetector, which reduces the parasitic capacitance of the diode. There is more leakage through the diode which causes a higher DC error.

USING THE FUNDAMENTALS

Instrumentation Amplifier

Instrumentation amplifiers are found in a large variety of applications from medical instrumentation to process control. The instrumentation amplifier is similar to the difference amplifier in that it subtracts one analog signal from another, but it differs in terms of the quality of the input stage. A classic, three op amp instrumentation amplifier is illustrated in Figure 9.

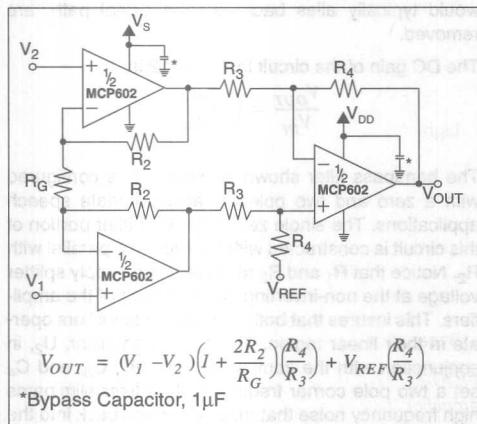


FIGURE 9: An instrumentation amplifier can be designed using three amplifiers. The input operational amplifiers provide signal gain. The output operational amplifier converts the signal from two inputs to a single ended output with a difference amplifier.

With this circuit the two input signals are presented to the high impedance non-inverting inputs of the amplifiers. This is a distinct advantage over the difference amplifier configuration when source impedances are high or mismatched. The first stage also gains the two incoming signals. This gain is simply adjusted with one resistor, R_G .

Following the first stage of this circuit is a difference amplifier. The function of this portion of the circuit is to reject the common mode voltage of the two input signals as well as differentiate them. The source impedances of the signals into the input of the difference amplifier are low, equivalent and well controlled.

The reference voltage of the difference stage of this instrumentation amplifier is capable of spanning a wide range. Most typically this node is referenced to half of the supply voltage in a signal supply application. A supply splitter such as the circuit in Figure 5 can be used for this purpose. The transfer function of this circuit is:

$$V_{OUT} = (V_1 - V_2) \left(I + \frac{2R_2}{R_G} \right) \left(\frac{R_4}{R_3} \right) + V_{REF} \left(\frac{R_4}{R_3} \right)$$

A second instrumentation amplifier is shown in Figure 10. In this circuit, the two amplifiers serve the functions of load isolation, and signal gain. The second amplifier also differentiates the two signals.

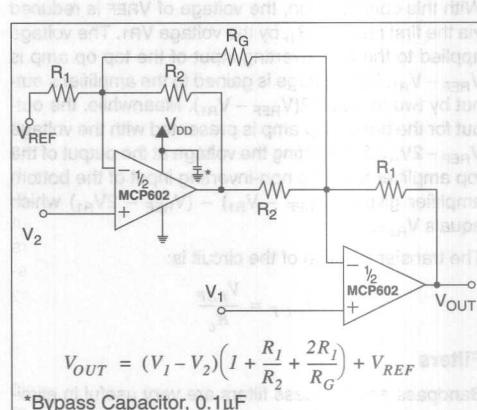


FIGURE 10: An instrumentation amplifier can be designed using two amplifiers. This configuration is best suited for higher gains. (gain $\geq 3 V/V$)

The circuit reference voltage is supplied to the first op amp in the signal chain. Typically, this voltage is half of the supply voltage in a single supply environment.

The transfer function of this circuit is:

$$V_{OUT} = (V_1 - V_2) \left(I + \frac{R_1}{R_2} + \frac{2R_1}{R_G} \right) + V_{REF}$$

Floating Current Source

A floating current source can come in handy when driving a variable resistance, like an Resistive Temperature Device (RTD). This particular configuration produces an appropriate 1mA source for an RTD type sensor, however, it can be tuned to any current.

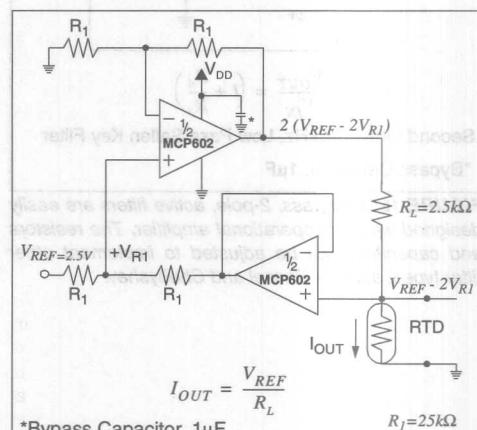


FIGURE 11: A floating current source can be constructed using two operational amplifiers and a precision voltage reference.

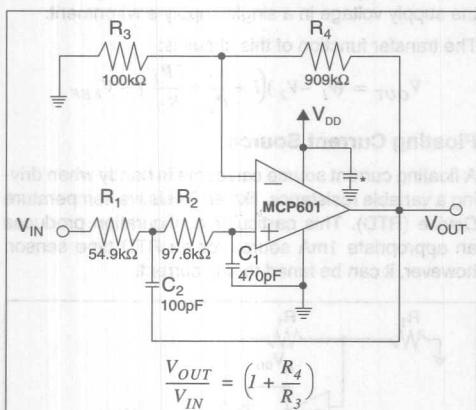
With this configuration, the voltage of V_{REF} is reduced via the first resistor, R₁, by the voltage V_{R1}. The voltage applied to the non-inverting input of the top op amp is V_{REF} - V_{R1}. This voltage is gained to the amplifier's output by two to equal 2(V_{REF} - V_{R1}). Meanwhile, the output for the bottom op amp is presented with the voltage V_{REF} - 2V_{R1}. Subtracting the voltage at the output of the top amplifier from the non-inverting input of the bottom amplifier gives 2(V_{REF} - V_{R1}) - (V_{REF} - 2V_{R1}) which equals V_{REF}.

The transfer function of the circuit is:

$$I_{OUT} = \frac{V_{REF}}{R_L}$$

Filters

Bandpass and low pass filters are very useful in eliminating unwanted signals prior to the input of an A/D converter. The low pass filter shown in Figure 12 has two poles that can be configured for a Butterworth filter response. Butterworth filters have a flat magnitude response in the pass-band with good all-around performance.



Second Order: 10kHz, Low Pass Sallen Key Filter

*Bypass Capacitor, 1μF

FIGURE 12: Low pass, 2-pole, active filters are easily designed with one operational amplifier. The resistors and capacitors can be adjusted to implement other filter types, such as Bessel and Chebyshev.

On the down side, there is some overshoot and ringing with a step response through this filter. This may or may not be an issue, depending on the application circuit requirements. The gain of this filter is adjustable with R₃ and R₄.

Notice the similarities in this gain equation and the non-inverting amplifier shown in Figure 3.

This type of filter is also referred to as an anti-aliasing filter, which is used to eliminate circuit noise in the frequency band above half of nyquist of the sampling system. In this manner, these high frequency noises, that would typically alias back into the signal path, are removed.

The DC gain of the circuit in Figure 12 is:

$$\frac{V_{OUT}}{V_{IN}} = \left(1 + \frac{R_4}{R_3}\right)$$

The bandpass filter shown in Figure 13 is configured with a zero and two poles to accommodate speech applications. The single zero high pass filter portion of this circuit is constructed with C₁ and R₁ in parallel with R₂. Notice that R₁ and R₂ also creates a supply splitter voltage at the non-inverting inputs of both of the amplifiers. This insures that both operational amplifiers operate in their linear region. The second amplifier, U₂, in conjunction with the components R₃, R₄, C₃, and C₄ set a two pole corner frequency. This filter eliminates high frequency noise that may be aliased back into the signal path.

The signal gain of this circuit is:

$$V_{OUT} = V_{IN} \left(\frac{R_3}{R_4} \right) \left(\frac{R_2}{R_1 + R_2} \right)$$

For more details about filters refer to AN699 "Anti-aliasing Analog Filters for Data Acquisitions Systems."

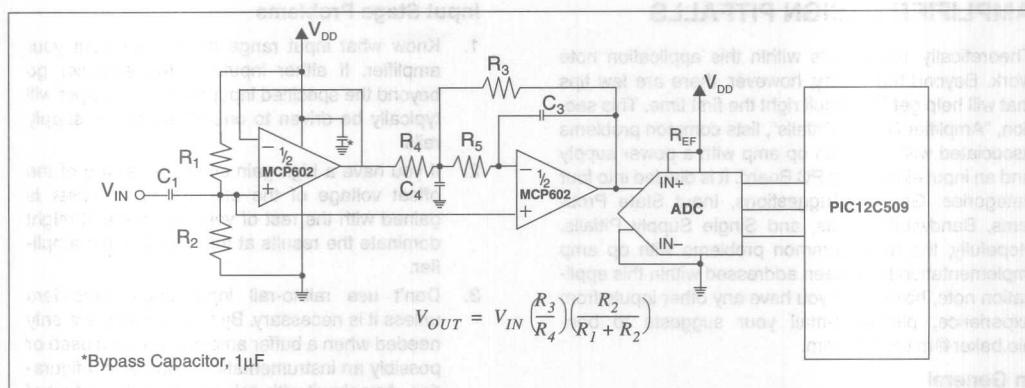


FIGURE 13: Band pass filters can be implemented with one operational amplifier designed to perform the high pass function and a second amplifier to perform the low pass function.

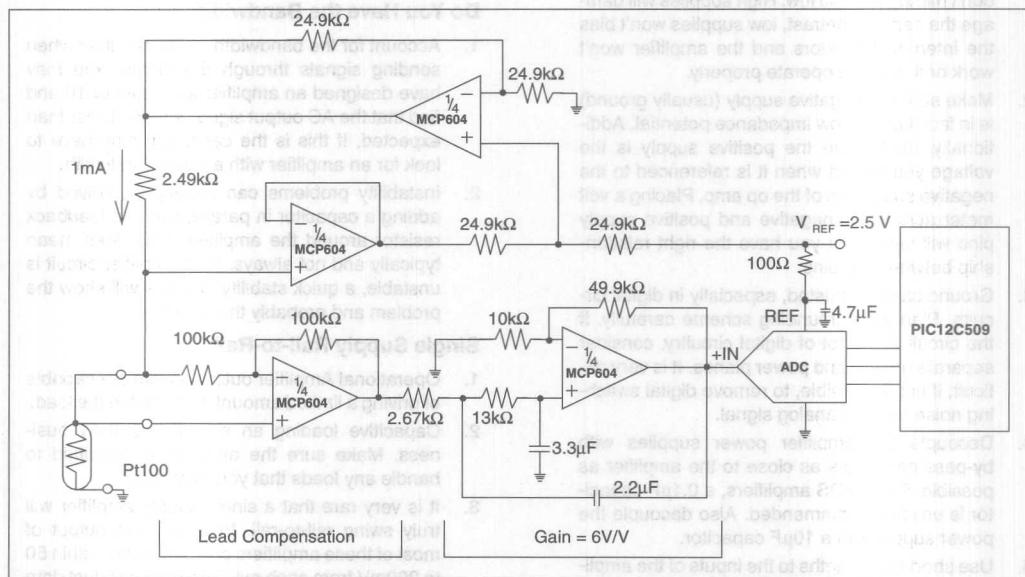


FIGURE 14: Complete single supply temperature measurement circuit.

Putting it Together

The circuit shown in Figure 14 utilizes four operational amplifiers along with a 12-bit A/D converter to implement a complete single supply temperature measurement circuit. The temperature sensor is an RTD which requires current excitation. The current excitation is supplied by the circuit described in Figure 11. The gain and anti-aliasing filter is implemented with the circuit shown in Figure 13.

The voltage signal from the RTD is sensed by an amplifier that is used in a combination of a non-inverting configuration and inverting configuration.

The output of this amplifier is then sent to an amplifier that is configured as a two pole, low pass filter in a gain of +6V/V. A gain of six was chosen in order to comply with the input range of A/D converter. Assuming the sampling frequency of the A/D converter is 75kHz, which is also known as the nyquist frequency, the cut-off frequency of the anti-aliasing filter (U4) is set to 10kHz. This allows plenty of bandwidth for the filter to attenuate the signal prior to 1/2 of nyquist. The A/D converter is a 12-bit Successive Approximation Register (SAR) converter that is interfaced to the PIC12C509 microcontroller.

AMPLIFIER DESIGN PITFALLS

Theoretically, the circuits within this application note work. Beyond the theory, however, there are few tips that will help get the circuit right the first time. This section, "Amplifier Design Pitfalls", lists common problems associated with using an op amp with a power supply and an input signal on a PC Board. It is divided into four categories: General Suggestions, Input Stage Problems, Bandwidth Issues, and Single Supply Pitfalls. Hopefully, the most common problems with op amp implementation have been addressed within this application note, however, if you have any other inputs from experience, please e-mail your suggests to bonnie.baker@microchip.com.

In General

1. Be careful of the supply pins. Don't make them too high per the amplifier specification sheet and don't make them too low. High supplies will damage the part. In contrast, low supplies won't bias the internal transistors and the amplifier won't work or it may not operate properly.
2. Make sure the negative supply (usually ground) is in fact tied to a low impedance potential. Additionally, make sure the positive supply is the voltage you expect when it is referenced to the negative supply pin of the op amp. Placing a voltmeter across the negative and positive supply pins will verify that you have the right relationship between the pins.
3. Ground can't be trusted, especially in digital circuits. Plan your grounding scheme carefully. If the circuit has a lot of digital circuitry, consider separate ground and power planes. It is very difficult, if not impossible, to remove digital switching noise from an analog signal.
4. Decouple the amplifier power supplies with by-pass capacitors as close to the amplifier as possible. For CMOS amplifiers, a $0.1\mu F$ capacitor is usually recommended. Also decouple the power supply with a $10\mu F$ capacitor.
5. Use short lead lengths to the inputs of the amplifier. If you have a tendency to use the white perf boards for prototyping, be aware that they can cause noise and oscillation. There is a good chance that these problems won't be a problem with the PCB implementation of the circuit.
6. Amplifiers are static sensitive! If they are damaged, they may fail immediately or exhibit a soft error (like offset voltage or input bias current changes) that will get worse over time.

Input Stage Problems

1. Know what input range is required from your amplifier. If either inputs of the amplifier go beyond the specified input range, the output will typically be driven to one of the power supply rails.
2. If you have a high gain circuit, be aware of the offset voltage of the amplifier. That offset is gained with the rest of your signal and it might dominate the results at the output of the amplifier.
3. Don't use rail-to-rail input stage amplifiers unless it is necessary. By the way, they are only needed when a buffer amplifier circuit is used or possibly an instrumentation amplifier configuration. Any circuit with gain will drive the output of the amplifier into the rail before the input has a problem.

Do You Have the Bandwidth?

1. Account for the bandwidth of the amplifier when sending signals through the circuit. You may have designed an amplifier for a gain of 10 and find that the AC output signal is much lower than expected. If this is the case, you may have to look for an amplifier with a wider bandwidth.
2. Instability problems can usually be solved by adding a capacitor in parallel with the feedback resistor around the amplifier. This does mean typically and not always. If an amplifier circuit is unstable, a quick stability analysis will show the problem and probably the solution.

Single Supply Rail-to-Rail

1. Operational Amplifier output drivers are capable of driving a limited amount of current to the load.
2. Capacitive loading an amplifier is risky business. Make sure the amplifier is specified to handle any loads that you may have.
3. It is very rare that a single supply amplifier will truly swing rail-to-rail. In reality, the output of most of these amplifiers can only come within 50 to 200mV from each rail. Check the product data sheets of your amplifier.

REFERENCES

- Sergio Franco, "Design with Operational Amplifiers and Analog Integrated Circuits", McGraw Hill
- Frederiksen, Thomas, "Intuitive Operational Amplifiers", McGraw Hill
- Williams, Jim, "Analog Circuit Design", Butterworth-Heinemann
- Baker, Bonnie, "Anti-aliasing Analog Filters for Data Acquisition Systems", AN699, Microchip Technology Inc.



MICROCHIP

AN684

Single Supply Temperature Sensing with Thermocouples

Author: *Bonnie C. Baker*
Microchip Technology Inc.

INTRODUCTION

There is a variety of temperature sensors on the market all of which meet specific application needs. The most common sensors used to solve these application problems include the thermocouple, Resistive Temperature Detector (RTD), Thermistor, and silicon based sensors. For an overview and comparison of these sensors, refer to Microchip's AN679, "Temperature Sensing Technologies".

This application note focuses on circuit solutions that use thermocouples in the design. The signal conditioning path for the thermocouple system will be discussed in this application note followed by complete application circuits.

THERMOCOUPLE OVERVIEW

Thermocouples are constructed of two dissimilar metals such as Chromel and Constantan (Type E) or Nicrosil and Nisil (Type N). The two dissimilar metals are bonded together on one end of both wires with a weld

bead. This bead is exposed to the thermal environment of interest. If there is a temperature difference between the bead and the other end of the thermocouple wires, a voltage will appear between the two wires at the end where the wires are not soldered together. This voltage is commonly called the thermocouple's Electromotive Force (EMF) voltage. This EMF voltage changes with temperature without any current or voltage excitation. If the difference in temperature between the two ends (the weld bead versus the unsoldered ends) of the thermocouple changes, the EMF voltage will change as well.

There are as many varieties of thermocouples as there are metals, but some combinations work better than others. The list of thermocouples shown in Table 1 are most typically used in industry. Their behaviors have been standardized by the National Institute of Standards and Technology (NIST). The particular document from this organization that is pertinent to thermocouples is the NIST Monograph175, "Temperature-Electromotive Force Reference Functions and Tables for the Letter-Designated Thermocouple Types Based on the ITS-90". Manufacturers use these standards to qualify the thermocouples that they ship.

2

**Application
Notes**

Thermocouple Type	Conductors	Temperature range (°C)	Seebeck Coefficient (@ 20°C)	Application Environments
E	Chromel (+) Constantan (-)	-200 to 900	62µV/°C	oxidizing, inert, vacuum
J	Iron (+) Constantan (-)	0 to 760	51µV/°C	vacuum, oxidizing reducing, inert
T	Copper (+) Constantan (-)	-200 to 371	40µV/°C	corrosive, moist, subzero
K	Chromel (+) Alumel (-)	-200 to 1260	40µV/°C	completely inert
N	Nicrosil (+) Nisil (-)	0 to 1260	27µV/°C	oxidizing
B	Platinum (30% Rhodium) (+) Platinum (6% Rhodium) (-)	0 to 1820	1µV/°C	oxidizing, inert
S	Platinum (10% Rhodium) (+) Platinum (-)	0 to 1480	7µV/°C	oxidizing, inert
R	Platinum (13% Rhodium) (+) Platinum (-)	0 to 1480	7µV/°C	oxidizing, inert

TABLE 1: Common thermocouple types—The most common thermocouple types are shown with their standardized material and performance specifications. These thermocouple types are fully characterized by the American Society for Testing and Materials (ASTM) and specified in IST-90 units per NIST Monograph 175.

Alico

Silicon sensors. As stated before, the sensor does not require any electrical excitation, such as a voltage or current source.

The price of thermocouples varies dependent on the purity of the metals, integrity of the weld bead and quality of the wire insulation. Regardless, thermocouples are relatively inexpensive as compared to other varieties of temperature sensors.

The thermocouple is one of the few sensors that can withstand hostile environments. The element is capable of maintaining its integrity over a wide temperature range as well as withstanding corrosive or toxic atmospheres. It is also resilient to rough handling. This is mostly a consequence of the heavier gages of wire used with the thermocouples construction.

The temperature ranges of the thermocouples included in Table 1 vary depending on the types of metals that are used. These ranges are also shown graphically in Figure 1. All of the voltages shown in Figure 1 are referenced to 0°C.

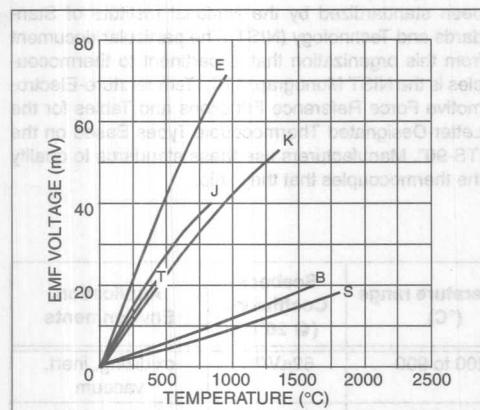


FIGURE 1: EMF voltage of various thermocouples versus temperature

micro volts to tens of millivolts. This voltage is repeatable, but non-linear. Although this can be seen to a certain degree in Figure 1, Figure 2 does a better job of illustrating the non-linearity of the thermocouple. In Figure 2, the first derivative of the EMF voltage versus temperature is shown. This first derivative at a specified temperature is called the Seebeck Coefficient. The Seebeck Coefficient is a linearized estimate of the temperature drift of the thermocouple's bead over a small temperature range. Since all thermocouples are non-linear, the value of this coefficient changes with specified temperature. This coefficient is used when designing the hardware portion of the thermocouple system that senses the absolute reference temperature. The design and use of the absolute temperature reference will be discussed later in this application note.

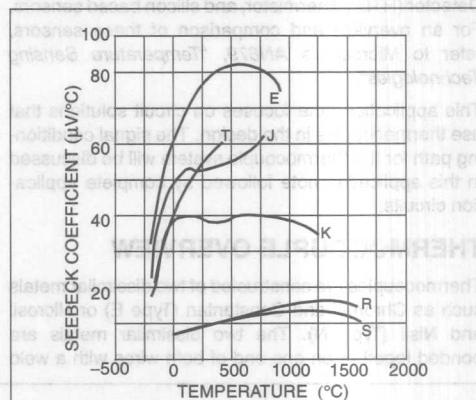


FIGURE 2: Seebeck coefficient of various thermo-couples versus temperature

From Figure 1, it can be summarized that the EMF voltage of a thermocouple is extremely small (millivolts). Additionally, Figure 2 illustrates that the change of the EMF voltage per degree C is also small ($\mu\text{V}/^\circ\text{C}$). Consequently, the signal conditioning portion of the electronics requires an analog gain stage. In addition, the voltage that a thermocouple produces represents the temperature difference between the weld bead and the other end of the wires. If an absolute temperature measurement (as opposed to relative) is required, a portion of the thermocouple signal conditioning electronics must be dedicated to establishing a temperature reference.

DESIGNING THE REFERENCE TEMPERATURE SENSOR

A summary of the thermocouple's advantages and disadvantages are listed in Table 2.

ADVANTAGES	DISADVANTAGES
No Excitation Required	Non-Linear
Inexpensive	Needs Absolute Temperature Reference
Wide Variety of Materials	Small Voltage Output Signals
Wide Temperature Ranges	
Very Rugged	

TABLE 2: Thermocouple Advantages and Disadvantages

THERMOCOUPLE SIGNAL CONDITIONING PATH

The signal conditioning signal path of the thermocouple circuit is illustrated in Figure 3. The elements of the path include the thermocouple, reference temperature junction, analog gain cell, Analog-to-Digital (A/D) Converter and the linearization block. Thermocouple 1 is the thermocouple that is at the site of the temperature measurement. Thermocouple 2 and 3 are a consequence of the wires of Thermocouple 1 connecting to the copper traces of the PCB.

The remainder of this application note will be devoted to solving the reference temperature, signal gain and A/D conversion issues. Linearization issues associated with thermocouples will also be discussed.

An absolute temperature reference is required in most thermocouple applications. This is used to remove the EMF error voltage that is created by thermocouples 2 and 3 in Figure 3. The two metals of these thermocouples come from the temperature sensing element (Thermocouple 1) and the copper traces of the PCB. The isothermal block in Figure 3 is constructed so that the Thermocouples 2 and 3 are kept at the same temperature as the absolute temperature sensing device. These elements can be kept at the same temperature by keeping the circuitry in a compact area, analyzing the board for possible hot spots, and identifying thermal hot spots in the equipment enclosure. With this configuration, the known temperature of the copper junctions can be used to determine the actual temperature of the thermocouple bead.

In Figure 3, the absolute reference temperature is sensed at the isothermal block, and then subtracted from the signal path. This is a hardware implementation. Alternatively, the absolute reference temperature can be sensed and subtracted in firmware. The hardware solution can be designed to be relatively error free as will be discussed later. The firmware correction can be more accurate because of the computing power of the processor. The trade-off for this type of calibration is computing time.

The relationship between the thermocouple bead temperature and zero degrees C is published in the form of look-up tables or coefficients of polynomials in the NIST publication mentioned earlier. If the absolute temperature of thermocouple 2 and 3 (Figure 3) are known, the actual temperature at the test sight (Thermocouple 1) can be measured and then calculated.

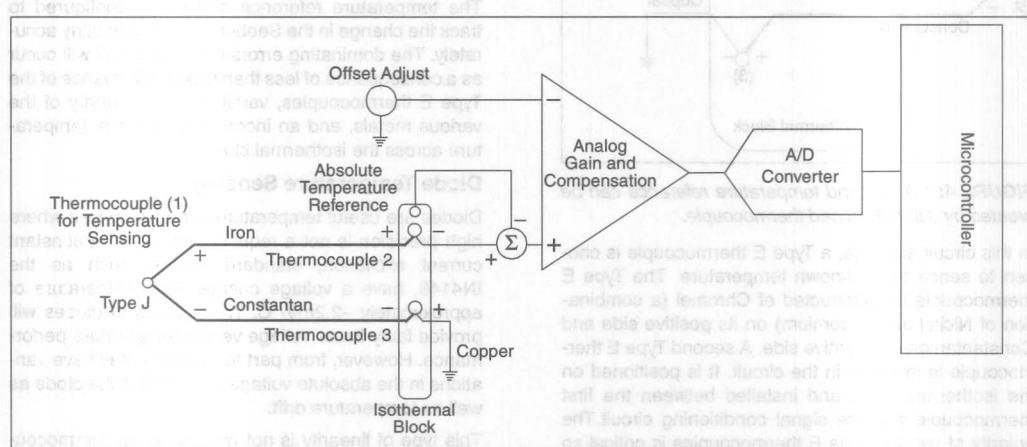


FIGURE 3: The thermocouple signal path starts with the thermocouple which is connected to the copper traces of the PCB on the isothermal block. The signal path then continues on to a differentiating circuit that subtracts the temperature of the isothermal block from the thermocouple's temperature. After this signal is digitized, a microcontroller uses the digital word from the temperature sensing circuit for further processing.

ERROR CORRECTION WITH HARDWARE IMPLEMENTATIONS

Many techniques can be used to sense the reference temperature on the isothermal block; five of which are discussed here. The first example uses a second thermocouple. It is used to sense ambient at the copper connection and configured to normalize the resultant voltage to an assignable temperature. As a second example, a standard diode is used to sense the absolute temperature of the isothermal block. This is done by using the negative temperature coefficient of $-2.2\text{mV}^{\circ}\text{C}$ characteristic of the diode. Thirdly, a thermistor temperature sensor is shown as the reference temperature device. As with the diode, the thermistor has a negative temperature coefficient. The thermistor is a more challenging to use because of its non-linear tendencies, however, the price is right. Another technique discusses an RTD as the reference temperature sensor. These sensors are best suited for precision circuits. And finally, the integrated silicon temperature sensor is briefly discussed.

Using a Second Thermocouple

A second thermocouple can be used to remove the error contribution of all of the thermocouples in the circuit. A circuit that uses this technique is shown in Figure 4.

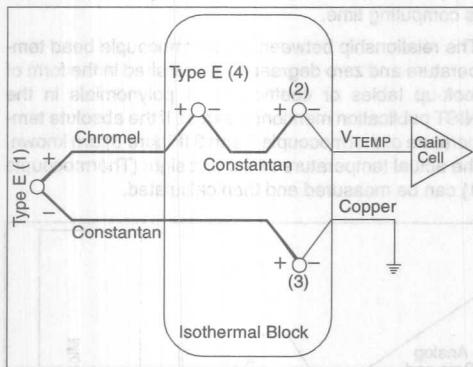


FIGURE 4: A second temperature reference can be created by using a second thermocouple.

In this circuit example, a Type E thermocouple is chosen to sense the unknown temperature. The Type E thermocouple is constructed of Chromel (a combination of Nickel and Chromium) on its positive side and Constantan on its negative side. A second Type E thermocouple is included in the circuit. It is positioned on the isothermal block and installed between the first thermocouple and the signal conditioning circuit. The polarity of the two Type E thermocouples is critical so that the Constantan on both of the thermocouples are connected together.

From this circuit configuration, two additional thermocouples are built, both of which are constructed with chromel and copper. These two thermocouples are opposing each other in the circuit. If both of these newly constructed thermocouples are at the same temperature, they will cancel each other's temperature induced errors.

The two remaining Type E thermocouples generate the appropriate EMF voltage that identifies the temperature at the sight of the first thermocouple.

This design technique is ideal for instances where the temperature of the isothermal block has large variations or the first derivative of voltage versus temperature of the selected thermocouple has a sharp slope (see Figure 2). Thermocouples that fit into this category in the temperature range from 0°C to 70°C are Type T and Type E.

The error calculation for this compensation scheme is:

$$V_{TEMP} = +EMF_3 + EMF_1 - EMF_4 - EMF_2$$

where

EMF_1 is the voltage drop across the Type E thermocouple at the test measurement site.

EMF_2 is the voltage drop across a Copper/Constantan thermocouple, where the copper metal is actually a PCB trace.

EMF_3 is the voltage drop across a Copper/Constantan thermocouple, where the copper metal is actually a PCB trace.

EMF_4 is the voltage drop across a Type E thermocouple on the Isothermal Block.

V_{TEMP} is the equivalent EMF voltage of a Type E thermocouple, #1, referenced to 0°C .

The temperature reference circuitry is configured to track the change in the Seebeck Coefficient fairly accurately. The dominating errors with this circuit will occur as a consequence of less than ideal performance of the Type E thermocouples, variations in the purity of the various metals, and an inconsistency in the temperature across the isothermal block.

Diode Temperature Sensing

Diodes are useful temperature sensing devices where high precision is not a requirement. Given a constant current excitation, standard diodes, such as the IN4148, have a voltage change with temperature of approximately $-2.2\text{mV}^{\circ}\text{C}$. These types of diodes will provide fairly linear voltage versus temperature performance. However, from part to part they may have variations in the absolute voltage drop across the diode as well as temperature drift.

This type of linearity is not well suited for thermocouples with wide variations in their Seebeck Coefficients over the temperature range of the isothermal block (referring to Figure 2). If there are wide variations with

the isothermal block temperature, Type K, J, R and S thermocouples may be best suited for the application. If the application requires more precision in terms of linearity and repeatability from part to part than an off-the-shelf diode, the MTS102, MTS103 or MTS105 from Motorola® can be substituted.

A circuit that uses a diode as an absolute temperature sensor is shown in Figure 5. A voltage reference is used in series with a resistor to excite the diode. The diode change with temperature has a negative coefficient, however, the magnitude of this change is much higher than the change of the collective thermocouple junctions on the isothermal block. This problem is solved by putting two series resistors in parallel with the diode. In this manner, the change of $-2.2\text{mV}^{\circ}\text{C}$ of the diode is attenuated to the Seebeck Coefficient of the thermocouple on the isothermal block. The Seebeck Coefficient of the thermocouples on the isothermal block are also equal to the Seebeck Coefficient (at isothermal block temperature) of the thermocouple that is being used at the test site. Table 3 has some recommended resistance values for various thermocouple types and excitation voltages.

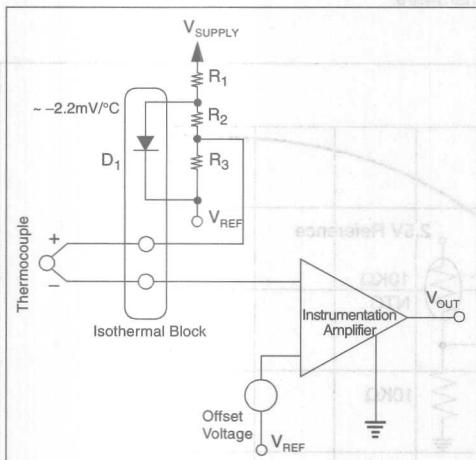


FIGURE 5: A diode can also be used in a hardware solution to zero out the temperature errors from the isothermal block.

This circuit appears to provide a voltage excitation for the diode. This is true, however, the ratio of the voltage excitation to the changes in voltage drop changes with temperature across the diode minimize linearity errors.

Of the three voltage references chosen in Table 3, the 10V reference provides the most linear results. It might also be noticed that changes in the reference voltage will also change the current through the diode. This being the case, a precision voltage reference is recommended for higher accuracy application requirements.

Thermocouple Type	Seebeck Coefficient ($\text{C} @ 20^{\circ}\text{C}$)	V_{REF} (V)	$R_1 (\Omega)$	$R_2 (\Omega)$	$R_3 (\Omega)$
J	$51\mu\text{V}^{\circ}\text{C}$	4.096	9.76k	4.22k	100
J	$51\mu\text{V}^{\circ}\text{C}$	5.0	12.1k	4.22k	100
J	$51\mu\text{V}^{\circ}\text{C}$	10.0	27k	4.22k	100
K	$40\mu\text{V}^{\circ}\text{C}$	4.096	9.76k	5.36k	100
K	$40\mu\text{V}^{\circ}\text{C}$	5.0	12.1k	5.36k	100
K	$40\mu\text{V}^{\circ}\text{C}$	10.0	27k	5.36k	100
R	$7\mu\text{V}^{\circ}\text{C}$	4.096	9.76k	31.6k	100
R	$7\mu\text{V}^{\circ}\text{C}$	5.0	12.1k	31.6k	100
R	$7\mu\text{V}^{\circ}\text{C}$	10.0	27k	31.6k	100
S	$7\mu\text{V}^{\circ}\text{C}$	4.096	9.76k	31.6k	100
S	$7\mu\text{V}^{\circ}\text{C}$	5.0	12.1k	31.6k	100
S	$7\mu\text{V}^{\circ}\text{C}$	10.0	27k	31.6k	100

TABLE 3: Recommended resistors and voltage references versus thermocouples for the circuit shown in Figure 5.

Thermistor Circuits

Thermistors are resistive devices that have a Negative Temperature Coefficient (NTC). These inexpensive sensors are ideal for moderate precision thermocouple sensing circuits when some or all of the non-linearity of the thermistor is removed from the equation.

The NTC thermistor's non-linearity can be calibrated out with firmware or hardware techniques. The firmware techniques are more accurate, however, hardware techniques are usually more than adequate. Details on these linearity issues of thermistors are discussed in Microchip's AN685, "Thermistors in Single Supply Temperature Sensing Circuits".

resistor and voltage excitation. In this circuit, the change in voltage with temperature is $\sim -25\text{mV}^{\circ}\text{C}$. This temperature coefficient is too high. A resistor divider (R_1 and R_2 in Figure 6) can easily provide the required temperature coefficient dependent on the thermocouple type.

This type of voltage excitation does have fairly linear operation over a limited temperature range (0°C to 50°C). Taking advantage of this linear region reduces firmware calibration overhead significantly.

Alternatively, the NTC thermistor can be excited with a current source. Low level current sources, such as $20\mu\text{A}$ are usually recommended which minimizes self heating problems. A thermistor that is operated with current firmwre excitation has a fairly non-linear output. With this type of circuit, firmware calibration would be needed. Although the firmware calibration is somewhat cumbersome, this type of excitation scheme can be more accurate.

Figure 7 compares the linearity of the thermistor with the current excitation configuration to a voltage excitation scheme shown in Figure 6.

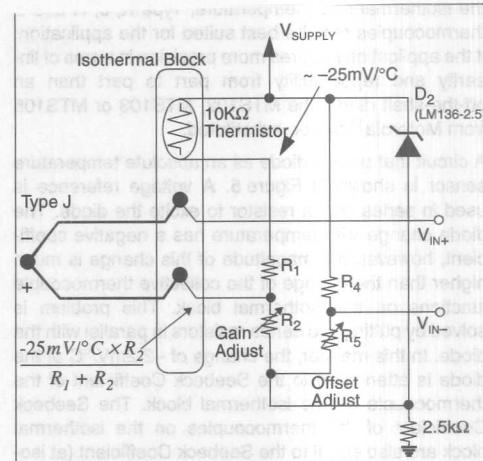


FIGURE 6: As a third method, a thermistor is used to sense the temperature of the isothermal block. In this circuit, the isothermal block error is eliminated in hardware.

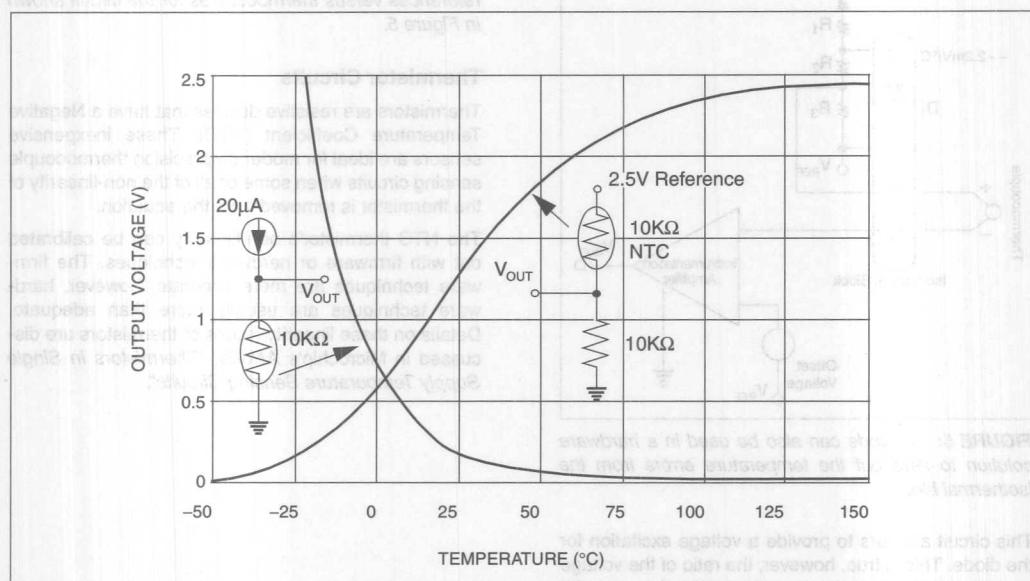


FIGURE 7: The Thermistor in Figure 6 requires linearization. This can be accomplished by using the Thermistor in series with a standard resistor.

RTD Sensor Circuits

Typically, an RTD would be used on the isothermal block if high precision is desired. The RTD element is nearly linear, consequently, employing linearization algorithms for the RTD is usually not required. The most effective way to get good performance from an RTD is to excite it with current. Both Figure 8 and Figure 9 show circuits that can be used for this purpose.

In Figure 8, a precision current reference is gained by the combination of R_1 , R_2 , J_1 , U_1 and U_2 . U_2 generates a $200\mu\text{A}$ precision current source. That current is pulled across R_1 forming a voltage drop for the power supply down to the non-inverting input of U_1 . U_1 is used to isolate R_1 from R_2 , while translating the voltage drop across R_1 to R_2 . In this manner, the $200\mu\text{A}$ current from U_2 is gained by the ratio of R_1/R_2 . J_1 is used to allow the voltage at the top of the RTD element to float dependent on its resistance changes with temperature. The RTD element should be sensed differentially. The voltage across this differential output is proportional to absolute temperature.

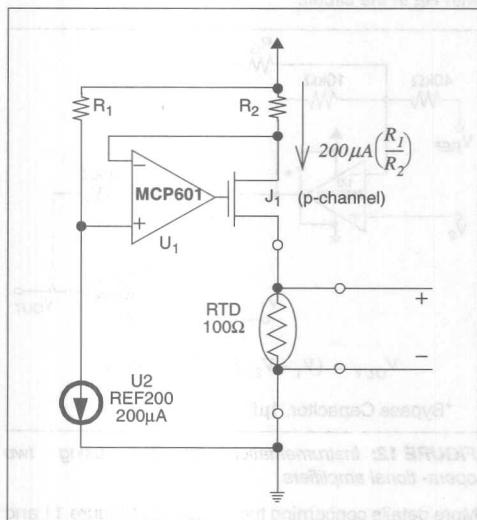


FIGURE 8: An 4-wire RTD can be used to sense the temperature of the isothermal block. RTDs require a precision current excitation as shown here.

In Figure 9, a voltage reference is used to generate a 1mA current source for the RTD element. The advantage of this configuration is that the voltage reference can be used elsewhere, allowing ratiometric calibration techniques in other areas of the circuit.

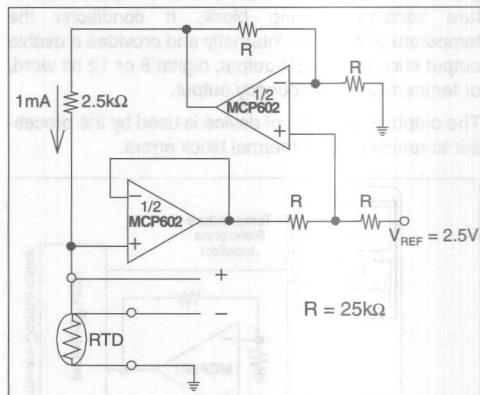


FIGURE 9: 3-wire RTD current excitation is generated with a precision voltage reference.

The RTD sensor is best suited for situations where precision is critical. Both of the RTD circuits (Figure 8 and Figure 9) will output a voltage that is fairly linear and proportional to temperature. This voltage is then used by the microcontroller to convert the absolute temperature reading of the isothermal block back to the equivalent EMF voltage. This can be performed by the microcontroller with a look-up table or a polynomial calculation for higher accuracy. This EMF voltage is then subtracted from the voltage measured across the sensor/isothermal block combination. In this manner, the errors from the temperature at the isothermal block are removed.

For more information about RTD circuits, refer to Microchip's AN687, "Precision Temperature Sensing with RTD Circuits".

Silicon Sensor

Silicon temperature sensors are differentiated from the simple diode because of their complexity (see Figure 10). A silicon temperature sensor is an integrated circuit that uses the diode as a basic temperature sensing building block. It conditions the temperature response internally and provides a usable output such as 0 to 5V output, digital 8 or 12 bit word, or temperature-to-frequency output.

The output of this type of device is used by the processor to remove the isothermal block errors.

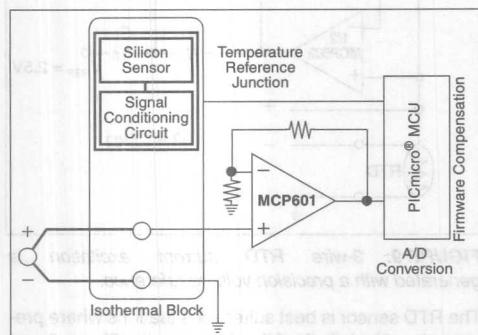


FIGURE 10: Silicon sensors are also useful for isothermal block temperature sensing. These type of devices only sense the temperature and do not implement any error correction in hardware.

SIGNAL CONDITIONING CIRCUITS

Once the reference temperature of the isothermal block is known, the temperature at the bead of the thermocouple can be determined. This is done by taking the EMF voltage, subtracting isothermal block errors, and determining the temperature through look-up tables or linearization equations. The EMF voltage must be digitized in order to easily perform these operations. Prior to the A/D conversion process, the low level voltage at the output of the thermocouple must be gained.

This is typically done with an instrumentation amplifier or a operational amplifier in a high gain configuration. An instrumentation amplifier uses several operational amplifiers and is configured to have a electrically equivalent differential inputs, high input impedance, potentially high gain, and good common-mode rejection. Of these four attributes, the first three are most useful for thermocouple applications.

Single supply configurations of instrumentation amplifiers are shown in Figure 11 and Figure 12. In Figure 11, three operation amplifier are used along with a selection of resistors. The circuit gain in Figure 11 can be controlled with R_G .

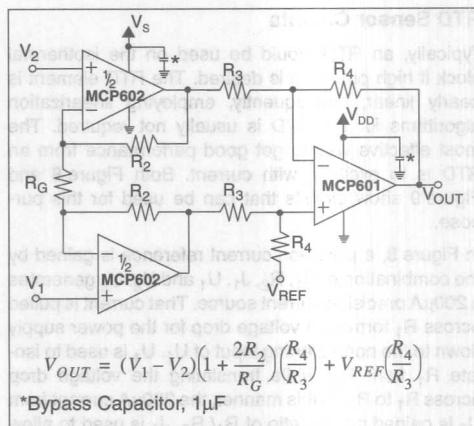


FIGURE 11: Instrumentation amplifier using three operational amplifiers

In Figure 12, an instrumentation amplifier is built using two amplifiers. Once again the gain is easily adjusted with R_G in the circuit.

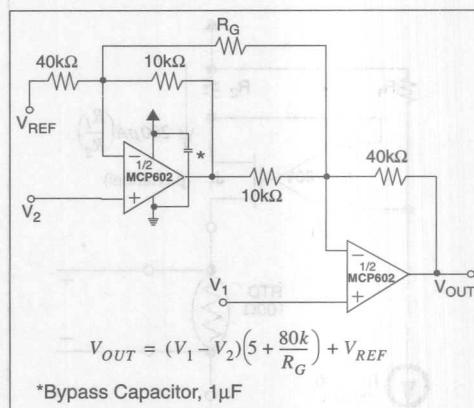


FIGURE 12: Instrumentation amplifier using two operational amplifiers

More details concerning the operation of Figure 11 and Figure 12 circuit configurations can be found in Microchip's AN682, "Using Single Supply Operational Amplifiers in Embedded Systems".

Finally, Figure 13 shows an circuit configuration using a single operational amplifier in an non-inverting gain.

These operational amplifier circuits will be used in the signal conditioning portion of the following thermocouple circuits.

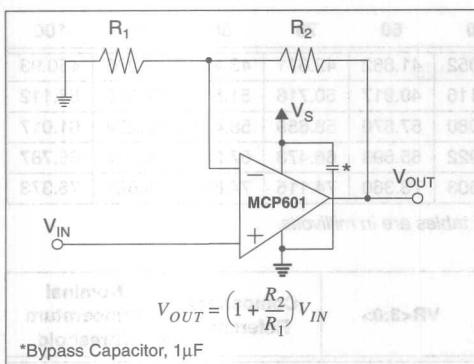


FIGURE 13: A single operational amplifier can be configured for analog gain.

THERMOCOUPLE CIRCUITS VERSUS ACCURACY

There are three types of thermocouple sensing systems in this section. The first circuit is designed to sense a threshold temperature. The second circuit will provide up to 8 bits of accuracy. This circuit accuracy can be improved by adding a higher resolution A/D Converter to the circuit, as shown in the third sensing system.

Threshold Temperature Sensing

A thermocouple can be used to sense threshold temperatures. This is particularly useful in industrial applications where high temperature processes need to be limited. The circuit to implement this type of function is shown in Figure 14. The threshold temperature sensing circuit in this figure combines the building blocks from Figure 4 and Figure 13.

This circuit is designed for simplicity. Consequently, all of the isothermal block error correction is performed in hardware. The Type E thermocouple is chosen for this circuit because of its high EMF voltage at high temperatures. This makes it easier to separate the real signal from background noise. Since the output of the isothermal block is single ended, the amplifier circuit in Figure 13 is used. In the event that there is a great deal of ambient or electrical noise, an instrumentation amplifier would serve this application better.

The EMF voltage of the thermocouple is calibrated across the isothermal block with a second thermocouple. This voltage is then gained by a single supply amplifier in a non-inverting configuration. The gain on the amplifier is adjustable by changing the ratio of R_2 and R_1 . In this case the signal is gained by 47.3V/V using a MCP601, single supply, CMOS operational amplifier. This gain was selected to provide a 2.5V output to the amplifier for a 700°C mid-scale measurement.

The microcontroller comparator can be programmed to compare between 1.25V and 3.75V with increments of $V_{DD}/32$ (LSB size of 156.25mV). This is done by configuring the CMCON register of the PIC16C62X to $CxOUT = 0$ and $CM<2:0 = 010$. Additionally, the voltage reference to the comparator is changed in the VRCON register. The initial settings for this register is $VREN = 1$ and $VRR = 0$. The processor can then cycle through the VRCON register $VR<3:0$ for a total of 16 different voltage reference settings for comparisons to the input signal from the MCP601 operational amplifier.

Temperature of interest ~700°C

Type E Chromel Constantan (2)

Type E Constantan Copper (3)

$R_1 = 432\Omega$

$R_2 = 20\Omega$

$V_{EMF} * \left(1 + \frac{R_2}{R_1}\right)$

MCP601

Comparator (4-bits, ranges from 1.25V to 3.75V)

PIC16C62X

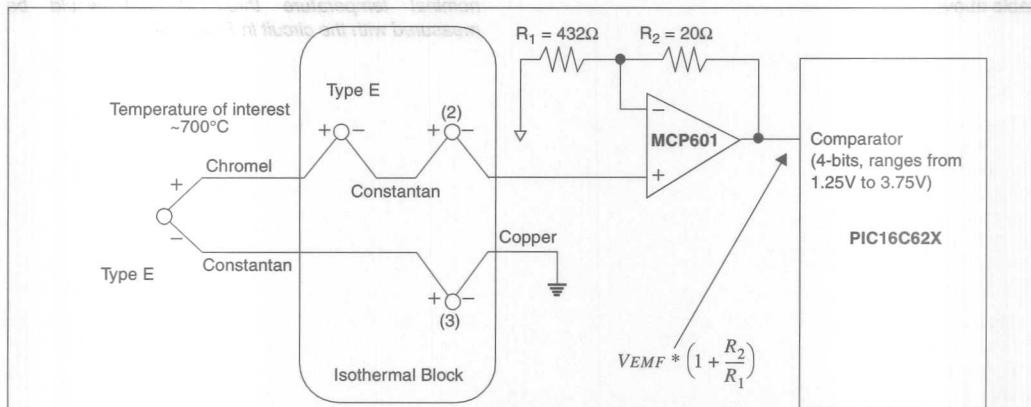


FIGURE 14: This circuit can be used to determine temperature thresholds. With calibration, the circuit is accurate to four bits.

$^{\circ}\text{C}$	0	10	20	30	40	50	60	70	80	90	100
500	37.005	37.815	38.624	39.434	40.243	41.052	41.862	42.671	43.479	44.286	450.93
600	45.093	45.900	46.705	47.509	48.313	49.116	49.917	50.718	51.517	52.315	53.112
700	53.112	53.908	54.709	55.497	56.289	57.080	57.870	58.659	58.446	60.232	61.017
800	61.017	61.801	62.583	63.364	64.144	64.922	65.698	66.473	67.246	68.017	68.787
900	68.787	69.554	70.319	71.082	71.844	72.603	73.360	74.115	74.869	75.621	76.373

TABLE 4: Type E thermocouple look-up table. All values in the tables are in millivolts.

A look-up table for the millivolts to 500°C to 1000°C for the Type E thermocouple is provided in Table 4. The temperature at the test sight is found by dividing the output voltage of the amplifier by 47.3 and using the look-up table to estimate the actual temperature. AN566, "Implementing a Table Read" can be used in this application to program the PICmicro® microcontroller.

Measurement errors (referred to the thermocouple) in this circuit come from, the offset voltage of the operational amplifier ($\pm 2\text{mV}$) and the comparator LSB size ($\pm 1.65\text{mV}$). Negligible error contributions come from the look-up table resolution, resistors and power supply variations.

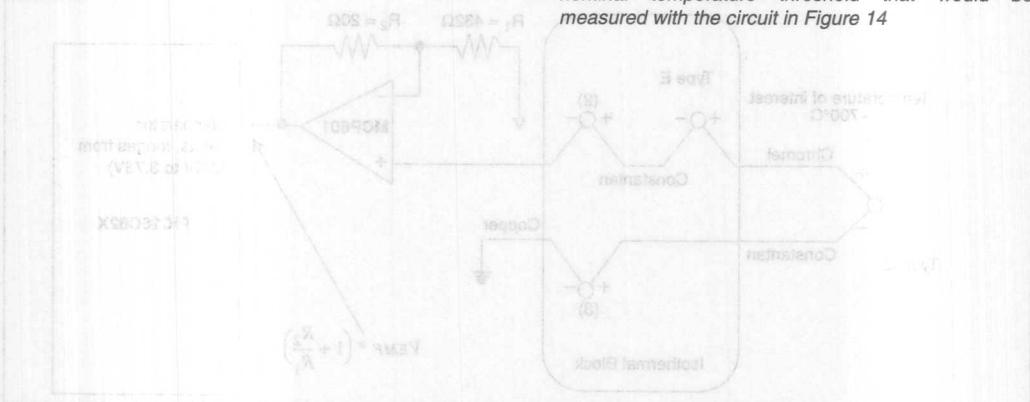
Given the errors above, the accuracy of the comparison in this circuit is $\sim \pm 35^{\circ}\text{C}$ over a nominal temperature range of 367.7°C to 992.6°C. This error can be calibrated out. The temperature thresholds for the various settings of VR<3:0> of the VRCON register is summarized in Table 5.

This accuracy can be improved by using an amplifier with less initial offset voltage or an A/D conversion with more bits.

All of the temperature calibration work in this circuit is performed in hardware. Linearization and temperature accuracy are performed in firmware with the look-up table above.

VR<3:0>	Comparator Reference	Nominal Temperature Threshold
0000	1.25V	368.4°C
0001	1.40625V	409.8°C
0010	1.5625V	450.9°C
0011	1.71875V	491.7°C
0100	1.875V	532.6°C
0101	2.03125V	573.4°C
0110	2.1875V	614.3°C
0111	2.34375V	655.4°C
1000	2.5V	696.8°C
1001	2.65625V	738.3°C
1010	2.8125V	780.2°C
1011	2.96875V	822.3°C
1100	3.125V	864.8°C
1101	3.28125V	907.6°C
1110	3.4375V	950.9°C
1111	3.59375V	994.7°C

TABLE 5: With a PIC16C62X controller, the comparator reference voltage is shown with the nominal temperature threshold that would be measured with the circuit in Figure 14.



Temperature Sensing up to 8-bits

An eight bit accurate thermocouple circuit is achievable by using the circuit shown in Figure 15. A Type K thermocouple is chosen for this circuit because of its stable Seebeck Coefficient between 0 and 50°C. Circuits from Figure 7 and Figure 11 are used to implement the reference temperature block as well as the signal conditioning block, respectively.

The thermistor is used as the absolute temperature sensor on the isothermal block. The combination of the thermistor and the surrounding resistors perform a first order linearization of the thermistor as discussed earlier.

The non-inverting input of the instrumentation amplifier (see Figure 11) is connected to the combination of the Type K thermocouple and the thermistor error correction circuitry. The inverting input of the instrumentation

amplifier is connected to the combination of R_4 and R_5 which provide an offset adjust capability. This offset adjustment capability is not needed if the temperature sensing application starts from 0°C. However, if the temperature of interest is above a certain threshold, the offset adjust can be used to improve the dynamic range of the measurement by allowing for the full-scale range of the instrumentation amplifier and the A/D Converter to be utilized.

Assuming that the temperature range of the measurement is from 500°C to 1000°C an appropriate offset voltage at the inverting input of the instrumentation amplifier would be 43.72mV for the combination of the Type K thermocouple offset at 750°C (per Table 6) and for the thermistor absolute temperature sensing circuit at 25°C.

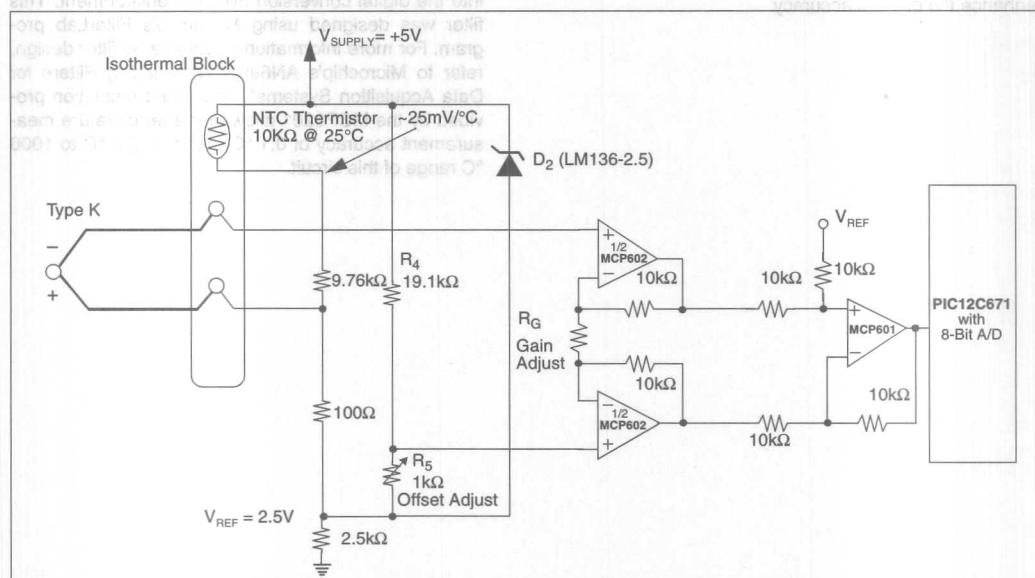


FIGURE 15: This circuit will provide 8-bit accurate temperature sensing results using a thermocouple. In this circuit, the A/D Converter is included in the PIC12C671 microcontroller.

°C	0	10	20	30	40	50	60	70	80	90	100
500	20.644	21.071	21.497	21.924	22.360	22.776	23.203	23.629	24.055	24.480	24.905
600	24.905	25.330	25.766	26.179	26.602	27.025	27.447	27.869	28.289	28.710	29.129
700	29.129	29.548	29.965	30.382	30.798	31.213	31.628	32.041	32.453	32.865	33.275
800	33.275	33.685	34.093	34.501	34.906	35.313	35.718	36.121	35.524	36.925	37.326
900	37.326	37.725	38.124	38.522	38.918	39.314	39.708	40.101	40.494	40.885	41.276

TABLE 6: Type K thermocouple output voltage look-up table. All values in the table are in millivolts.

Assuming that the offset has been minimized, the output range of the thermocouple circuit for an excursion from 500°C to 1000°C is $\Delta 20.632\text{mV}$.

The output of the instrumentation amplifier swings up to $V_{DD} - 100\text{mV}$. In this single supply, 5V environment, the output of the MCP601 operational amplifier will swing from 100mV to 4.9V.

The differential voltage swing at the inputs to the instrumentation amplifier is -17.41mV to $+16.13\text{mV}$ centered around the voltage reference of 2.5V.

Given a full-scale voltage of 33.54mV from the temperature sensing circuit, the instrumentation amplifier can be configured for a gain of 137.85V/V. This gain can easily be implemented by making R_G equal to 147Ω . This circuit is not restricted to 8-bits of accuracy. An external A/D Converter such as one of Microchip's 12-bit A/D Converter, MCP320X, can be used to further enhance the circuit's accuracy.

High Precision Temperature Sensing with a 12-bit Converter

The circuit shown in Figure 15 can be further enhanced to allow for 12-bit accuracy with the addition of a MCP3201 12-bit A/D Converter and a 4th order low pass analog filter. With this circuit, the PIC12C671 is replaced with the PIC12C509.

The analog circuit in Figure 16, remains unchanged from the design shown in Figure 15 up to the analog low pass filter. This additional low pass filter is constructed using the MCP602, CMOS dual operational amplifier. The 4th order low pass filter Butterworth design that is implemented in this circuit has a cut-off frequency of 10Hz. This cut-off frequency assumes that the sample rate of the MCP3201 is 20Hz or greater. The analog filter is used to remove the instrumentation amplifier noise, as well as the noise that may be aliased into the digital conversion from the environment. This filter was designed using Microchip's FilterLab program. For more information about analog filter design, refer to Microchip's AN699, "Anti-Aliasing Filters for Data Acquisition Systems". The 12-bit resolution provided by the MCP3201 allows for a temperature measurement accuracy of 0.1 °C over the 500 °C to 1000 °C range of this circuit.

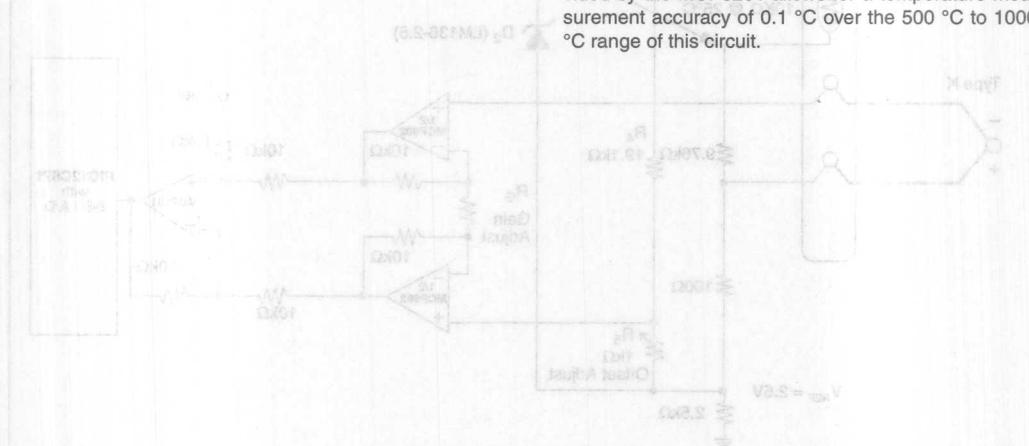


Figure 16: High Precision Temperature Sensing with a 12-bit Converter

0.0	0.5	1.0	0.5	0.0	0.5	1.0	0.5	0.0	0.5	1.0	0.5	0.0	0.5	1.0	0.5	0.0	0.5	1.0	0.5	0.0	0.5	1.0	0.5	0.0	
300.0S	304.4S	308.8S	313.2S	317.6S	322.0S	326.4S	330.8S	335.2S	340.6S	345.0S	349.4S	353.8S	358.2S	362.6S	367.0S	371.4S	375.8S	380.2S	384.6S	389.0S	393.4S	397.8S	402.2S	406.6S	
301.0S	305.3S	309.6S	313.9S	318.2S	322.5S	326.8S	331.1S	335.4S	339.7S	344.0S	348.3S	352.6S	356.9S	361.2S	365.5S	369.8S	374.1S	378.4S	382.7S	387.0S	391.3S	395.6S	399.9S	404.2S	408.5S
302.0S	306.3S	310.6S	314.9S	319.2S	323.5S	327.8S	332.1S	336.4S	340.7S	345.0S	349.3S	353.6S	357.9S	362.2S	366.5S	370.8S	375.1S	379.4S	383.7S	388.0S	392.3S	396.6S	400.9S	405.2S	409.5S
303.0S	307.3S	311.6S	315.9S	320.2S	324.5S	328.8S	333.1S	337.4S	341.7S	346.0S	350.3S	354.6S	358.9S	363.2S	367.5S	371.8S	376.1S	380.4S	384.7S	389.0S	393.3S	397.6S	401.9S	406.2S	410.5S
304.0S	308.3S	312.6S	316.9S	321.2S	325.5S	329.8S	334.1S	338.4S	342.7S	347.0S	351.3S	355.6S	359.9S	364.2S	368.5S	372.8S	377.1S	381.4S	385.7S	390.0S	394.3S	398.6S	402.9S	407.2S	411.5S

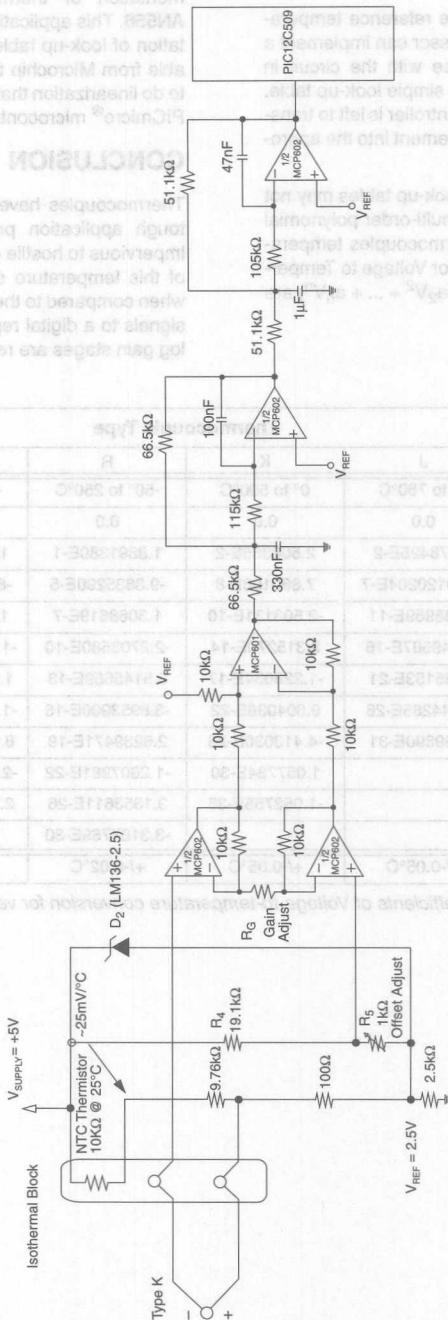


FIGURE 16: This circuit will provide 12-bit accurate temperature sensing results using a thermocouple. In this circuit, an external A/D Converter (MCP3021), is used to digitize the analog signal.

THERMOCOUPLE LINEARIZATION

Once a voltage from the absolute reference temperature sensor is digitized, the processor can implement a variety of algorithms. In the case with the circuit in Figure 15, the processor scans a simple look-up table. With this type of data, the microcontroller is left to translate the signal from the sensing element into the appropriate EMF voltage.

For high precision applications, look-up tables may not be adequate. In these cases, a multi-order polynomial can be used to generate the thermocouples temperature. The polynomial coefficients for Voltage to Temperature Conversion ($T = a_0 + a_1V + a_2V^2 + \dots + a_nV^n$) are shown in Table 7.

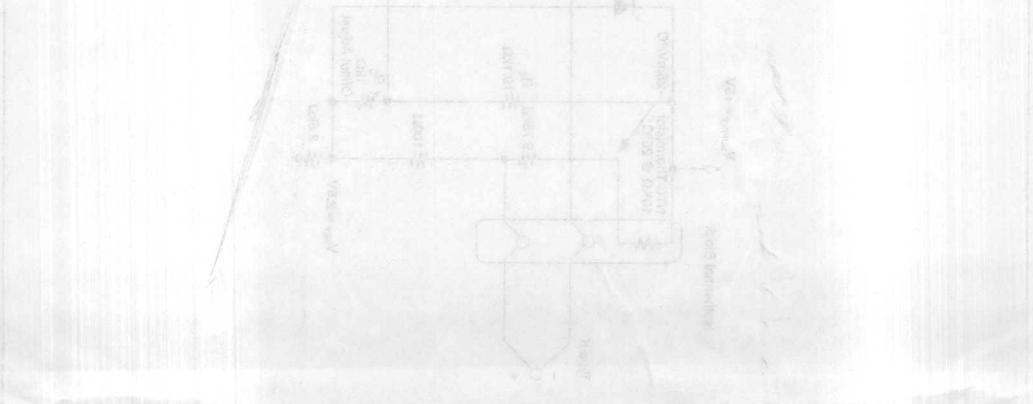
For further discussion concerning the firmware implementation of thermocouple linearization, refer to AN556. This application note discussed the implementation of look-up tables. Additionally, firmware is available from Microchip that provides look-up tables code to do linearization that is directly programmable into the PICmicro® microcontroller of your choice.

CONCLUSION

Thermocouples have their advantages when used in tough application problems. They are rugged and impervious to hostile environments. The voltage output of this temperature sensing element is relatively low when compared to the devices that can convert voltage signals to a digital representation. Consequently, analog gain stages are required in the circuit.

	Thermocouple Type					
	E	J	K	R	S	T
Range	0° to 1000°C	0° to 760°C	0° to 500°C	-50° to 250°C	-50° to 250°C	0° to 400°C
a ₀	0.0	0.0	0.0	0.0	0.0	0.0
a ₁	1.7057035E-2	1.978425E-2	2.508355E-2	1.8891380E-1	1.84949460E-1	2.592800E-2
a ₂	-2.3301759E-7	-2.00120204E-7	7.860106E-8	-9.3835290E-5	-8.00504062E-5	-7.602961E-7
a ₃	6.543558E-12	1.036969E-11	-2.503131E-10	1.3068619E-7	1.02237430E-7	4.637791E-11
a ₄	-7.3562749E-17	-2.549687E-16	8.315270E-14	-2.2703580E-10	-1.52248592E-10	-2.165394E-15
a ₅	-1.7896001E-21	3.585153E-21	-1.228034E-17	3.5145659E-13	1.88821343E-13	6.048144E-20
a ₆	8.4036165E-26	-5.344285E-26	9.804036E-22	-3.8953900E-16	-1.59085941E-16	-7.293422E-25
a ₇	-1.3735879E-30	5.099890E-31	-4.413030E-26	2.8239471E-19	8.23027880E-20	
a ₈	1.0629823E-35		1.057734E-30	-1.2607281E-22	-2.34181944E-23	
a ₉	-3.2447087E-41		-1.052755E-35	3.1353611E-26	2.79786260E-27	
a ₁₀				-3.3187769E-30		
Error	+/-0.02°C	+/-0.05°C	+/-0.05°C	+/-0.02°C	+/-0.02°C	+/-0.03°C

TABLE 7: NIST Polynomial Coefficients of Voltage-to-temperature conversion for various thermocouple type



REFERENCES

- Baker, Bonnie, "Thermistors in Single Supply Temperature Sensing Circuits", *AN685, Microchip Technology Inc.*, 1998
- Baker, Bonnie, "Precision Temperature Sensing with RTD Circuits", *AN687, Microchip Technology Inc.*, 1998
- Baker, Bonnie, "Temperature Sensing Technologies", *AN679, Microchip Technology Inc.*, 1998
- Baker, Bonnie, "Anti-Aliasing Filters for Data Acquisition Systems", *AN699, Microchip Technology Inc.*, 1998
- Klopfenstein, Rex, "Software Linearization of a Thermocouple", *SENSORS*, Dec. 1997, pg 40
- "Practical Temperature Measurements", *OMEGA Catalog*, pg 2-11
- "Thermocouples and Accessories", *Measurement & Control*, June 1996, pg 190
- "RTD Versus Thermocouple", *Measurement & Control*, Feb., 1997, pg 108
- "Ya Can't Calibrate a Thermocouple Junction!, Part 2 - So What?" *Measurement & Control*, Oct. 1996, pg 93
- "A Comparison of Programs That Convert Thermocouple Properties to the 1990 International Temperature & Voltage Scales", *Measurement & Control*, June, 1996, pg 104
- "Thermocouple Basics", *Measurement & Control*, June, 1996, pg 126
- G.W. Burns, M.G. Scroger, G.F. Strouse, et al. Temperature-Electromotive Force Reference Functions and Tables for the Letter-Designated Thermocouple Types Based on the IPTS-90 NIST Monograph 175. Washington, D.C.: U.S. Department of Commerce, 1993
- D'Sousa, Stan, "Implementing a Table Read", *AN556, Microchip Technology Inc.*, 1997

NOTES:





MICROCHIP

AN685

Thermistors in Single Supply Temperature Sensing Circuits

Author: **Bonnie C. Baker**
Microchip Technology Inc.

INTRODUCTION

There is a variety of temperature sensors on the market all of which meet specific application needs. The most common sensors that are used to solve these application problems include the thermocouple, Resistive Temperature Detector (RTD) thermistor, and silicon-based sensors. For an overview and comparison of these sensors, refer to Microchip's AN679, "Temperature Sensing Technologies".

This application note focuses on circuit solutions that use Negative Temperature Coefficient (NTC) thermistors in the design. The Thermistor has a non-linear resistance change-over temperature. The degree of this non-linearity will be discussed in the "Hardware Linearization Solutions" section of this application note. From this discussion, various linearization resistor networks will be shown with error analysis included. Finally, the signal conditioning path for the thermistor system will be covered with complete application circuits from sensor or microprocessor.

THERMISTOR OVERVIEW

The term "thermistor" originated from the descriptor THERMally Sensitive Resistor. The two basic types of thermistors are the Negative Temperature Coefficient (NTC) and Positive Temperature Coefficient (PTC). The NTC thermistor is best suited for precision temperature measurement. The PTC is best suited for switching applications. This application note will only discuss NTC applications.

The NTC thermistor is used in three different modes of operation which services a variety of applications. One of the modes exploits the resistance-versus-temperature characteristics of the thermistor. The other two modes take advantage of the voltage-versus-current and current-over-time characteristics of the thermistor.

Voltage-Versus-Current Mode

Voltage-versus-current applications use one or more thermistors that are operated in a self-heated, steady-state condition. An application example for an NTC thermistor in this state of operation would be using a flow meter. In this type of circuit, the thermistor would be in an ambient self-heated condition. The ther-

mistor's resistance is changed by the amount of heat generated by the power dissipated by the element. Any change in the flow of the liquid or gas across the device changes the power dissipation factor of the thermistor element. In this manner, the resistance of the thermistor is changed, relative to the degree of cooling provided by the flow of liquid or gas. A useful thermistor graph for this phenomena is shown in Figure 1. The small size of the thermistor allows for this type of application to be implemented with minimal interference to the system. Applications such as vacuum manometers, anemometers, liquid level control, fluid velocity and gas detection are used with the thermistors in voltage-versus-current mode.

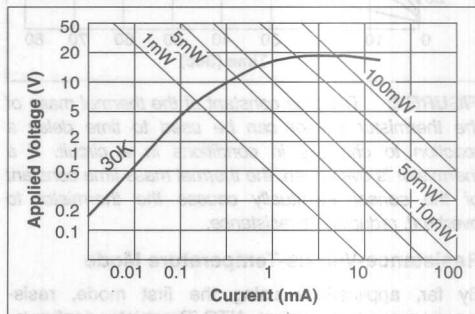


FIGURE 1: When a thermistor is overheated by its own power, the device operates in the voltage-versus-current mode. In this mode, the thermistor is best suited to sense changes in the ambient conditions, such as changes in the velocity of air flow across the sensor.

Current-Over-Time Mode

The current-over-time characteristics of a thermistor also depends on the dissipation constant of the thermistor package as well as element's heat capacity. As current is applied to a thermistor, the package will begin to self-heat. If the current is continuous, the resistance of the thermistor will start to lessen. The thermistor current-time characteristics can be used to slow down the affects of a high voltage spike, which could be for a short duration. In this manner, a time delay from the thermistor is used to prevent false triggering of relays.

The effect of the thermistor current-over-time delay is shown in Figure 2. This type of time response is relatively fast as compared to diodes or silicon based temperature sensors. The diode and silicon based sensors require several minutes to reach their steady state temperature. In contrast, thermocouples and RTDs are equally as fast as the thermistor, but they don't have the equivalent high level outputs. Applications based on current-over-time characteristics include time delay devices, sequential switching, surge suppression or inrush current limiting.

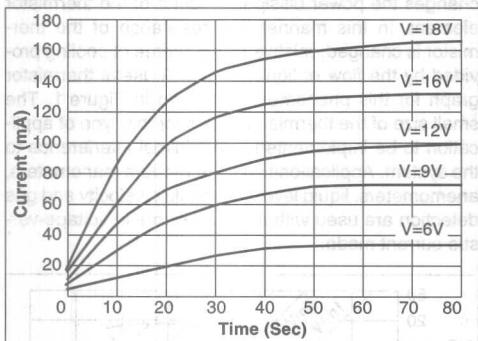


FIGURE 2: The time constant of the thermal mass of the thermistor sensor can be used to time delay a reaction to changes in conditions in a circuit. If a thermistor is overdriven, the thermal mass time constant of the sensor eventually causes the thermistor to overheat, reducing its resistance.

Resistance-Versus-Temperature Mode

By far, applications using the first mode, resistance-versus-temperature, NTC Thermistor configurations, are the most prevalent. These circuits perform precision temperature measurement, control and compensation. Unlike applications that are based on the voltage-versus-current and current-over-time characteristics of the thermistor, the resistance-versus-temperature circuits depend on the thermistor being operated in a "zero-power" condition. This condition implies that there is no self-heating of the thermistor as a consequence of current or voltage excitation. The resistance-versus-temperature response of a $10\text{k}\Omega$ NTC thermistor is shown in Figure 3.

The resistance across the thermistor is relatively high in comparison to the RTD element which is usually in the hundreds of ohms range. Typically, the 25°C rating for thermistors is from $1\text{k}\Omega$ up to $10\text{M}\Omega$. The housing of the thermistor varies as the requirements for hermeticity and ruggedness vary, but in all cases, there are only two wires going to the element. This is possible because of the resistance of the wiring over temperature is considerably lower than the thermistor element. Consequently, a four wire configuration is not necessary, as it is with the RTD element. (Refer to AN687, "RTD Temperature Sensing Circuits" for details.)

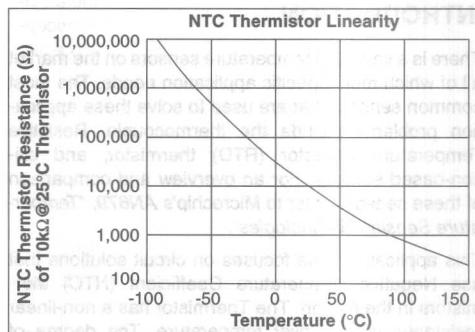


FIGURE 3: In precision temperature measurement environments, the thermistor is used in a "zero power" condition. In this condition, the power consumption of the thermistor has a negligible affect on the elements resistance. This is a graph of an NTC $10\text{k}\Omega$ thermistor resistance-versus-temperature.

Since the thermistor is a resistive element, current excitation is required. The current can originate from a voltage or current reference, as will be shown in the "Hardware Linearization Solutions" section of this application note. The performance of the thermistor in Figure 3 is fairly repeatable as long as the power across the device does not exceed the power dissipation capability of the package. Once this condition is violated, the thermistor will self-heat and artificially decrease in resistance, giving a higher than actual temperature reading.

To avoid thermal self-heating, the power dissipation of the thermistor must be considered. The power dissipation is calculated by the formula $P = IV$, where I is the current through the thermistor and V is the voltage across the thermistor. The power dissipation is proportional to the square of the current and the voltage across the thermistor. As the current increases, the power dissipation increases, causing the thermistor to heat up. This heating can cause a change in the resistance of the thermistor, which can lead to inaccurate temperature measurements.

Example Thermistor at 25°C: 10kΩ

For example, if a 10kΩ thermistor is connected to a 5V DC power source and a 1mA current flows through it, the power dissipation would be $P = 5 \text{V} \times 0.001 \text{A} = 0.005 \text{W}$. This is a very small amount of power, but it is enough to cause a significant increase in temperature if the thermistor is not properly cooled. The temperature rise can be calculated using the formula $\Delta T = P / R$, where P is the power dissipation and R is the resistance of the thermistor. In this case, the temperature rise would be $\Delta T = 0.005 \text{W} / 10\text{k}\Omega = 0.5^\circ\text{C}$.

Figure 3 illustrates the high degree of non-linearity of the thermistor element. Although the thermistor has considerably better linearity than the thermocouple linearity, the thermistor still requires linearization in most temperature sensing circuits. The non-linear response of the thermistor can be corrected in software with an empirical third-order polynomial or a look-up table. There are also easy hardware linearization techniques that can be applied prior to digitalization of the output of the thermistor. These techniques will be discussed later in this application note. The third-order polynomial is also called the Steinhart-Hart Thermistor equation. This equation is an approximation and can replace the exponential expression for a thermistor. Wide industry acceptance makes it the most useful equation for precise thermistor computation.

The Steinhart-Hart equation is:

$$T = 1/(A_0 + A_1(\ln R_T) + A_3(\ln R_T)^3)$$

$$\ln R_T = B_0 + B_1/T + B_3/T^3$$

where:

T is the temperature of the thermistor in Kelvin.

A_0, A_1, A_3, B_0, B_1 , and B_3 , are constants provided by the thermistor manufacturer.

R_T is the thermocouple resistance at temperature, T .

With a typical thermistor, this third-order linearization formula provides $\pm 0.1^\circ\text{C}$ accuracy over the full temperature range. This is usually better than the accuracy of individual elements from part to part.

Although the temperature range of the thermistor is a little better than the diode or silicon-based temperature sensor (-55°C to $+175^\circ\text{C}$), it is still limited to a practical range of -100°C to $+175^\circ\text{C}$. This can also be compared to the temperature sensing range of the RTD (-200°C to 600°C) or the thermocouple which ranges up to 1820°C .

The advantages versus disadvantages of the thermistor are summarized in Table 1.

ADVANTAGES	DISADVANTAGES
Fast	Non-Linear
Small	Excitation Required
Two-Wire	Limited Temperature Range
Inexpensive	Self-Heating
	Fragile

TABLE 1: Summary of Thermistor Advantages and Disadvantages.

Thermistors are manufactured by a large variety of vendors. Each vendor carefully specifies their thermistor characteristics with temperature, depending on their manufacturing process. Of all of the temperature sensors, the thermistor is the least expensive sensing element on the market. Prices start at \$0.10 with some vendors and range up to \$25.

The thermistor is used in a large variety of applications such as automotive monitor and control exhaust emissions, ice detection, skin sensors, blood and urine analyzers, refrigerators, freezers, mobile phones, base stations laser drives, and battery pack charging. In the precision instrumentation applications, thermistors are used in hand-held meters and temperature gauges.

Although the temperature range of the thermistor is a little better than the diode or silicon-based temperature sensor (-55°C to $+175^\circ\text{C}$), it is still limited to a practical range of -100°C to $+175^\circ\text{C}$. This can also be compared to the temperature sensing range of the RTD (-200°C to 600°C) or the thermocouple which ranges up to 1820°C .

THE TEMPERATURE- RESISTIVE MODE OF THE THERMISTOR

An electrical configuration for the thermistor is shown in Figure 4. This illustrates a seemingly obvious way to excite the thermistor and measure the change in resistance where the sensing element is excited with a current source.

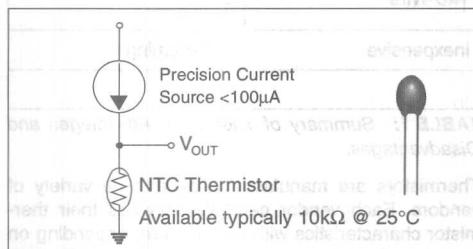


FIGURE 4: Common sense would dictate that the thermistor be excited by a precision constant current source as shown in this figure. A picture of an NTC Thermistor is shown on the right.

With this style of excitation, the magnitude of the current source is typically below 100µA, preferably 20µA. Lower currents prevent the thermistor from entering a self-heating condition as described previously. This style of excitation is effective for sensing a limited range of temperatures. Larger ranges of temperature have deltas in resistance that are too high to accurately convert the resistance to voltage without bumping into the noise limitations of the analog signal path.

As an example, the temperature range of a typical thermistor from BetaTHERM is -80°C to 150°C. The change in resistance for a 10kΩ @ 25°C thermistor from BetaTHERM over its temperature range is shown in Table 2.

It is useful to note that the differential resistance for a 10°C delta at high temperature is significantly smaller than a 10°C delta at low temperatures. For instance, the change in resistance of the device in Table 2 from 125°C to 135°C is 76.28Ω (340.82Ω – 264.54Ω). The change in resistance of the same thermistor from -25°C to -15°C is 58.148kΩ. This diversity in the ratio of resistance to temperature over the range of thermistor creates an awkward analog problem. If the thermistor in this example is excited with a 20µA current source, the analog circuit must discriminate between 0.015V deltas at high temperatures and 1.16V deltas at low temperatures for Δ10°C of resolution. This forces the LSB size in a linear digitizing system to be 1/2 of 0.015V. This would require a 9.57-bit system to achieve 10°C accuracy from the system over a temperature span of -25°C to 135°C (delta of 160°C).

Temp (°C)	R Value (Ω)	Temp (°C)	R Value (Ω)	Temp (°C)	R Value (Ω)
-80	7296874	0	32650.8	75	1480.12
-75	4713762	5	253985.5	80	1256.17
-70	3095611	10	19903.5	85	1070.58
-65	2064919	15	15714.0	90	916.11
-60	1397935	20	12493.7	95	786.99
-55	959789	25	10000	100	678.63
-50	667828	30	8056.0	105	587.31
-45	470609	35	6530.1	110	510.06
-40	335671	40	5324.9	115	44.48
-35	242195	45	4366.9	120	388.59
-30	176683	50	3601.0	125	340.82
-25	130243	55	2985.1	130	299.82
-20	96974	60	2487.1	135	264.54
-15	72895	65	2082.3	140	234.08
-10	55298	70	1751.6	145	207.70
-5	42314.6			150	184.79

TABLE 2: Resistive changes with temperature of a BetaTHERM, 10kΩ @ 25°C (10K3A1) NTC Thermistor in its "zero power" mode.

LINEARIZATION SOLUTIONS

It is obvious in this example that the conversion process is inefficient if a linear response is required. It is also obvious that the digital output word will require a look-up table to linearize the response. Additionally, temperature accuracy is usually required for most systems. These problems can be solved to a small degree by using a high resolution Analog-To-Digital (A/D) Converting device. In this scenario, bits will still be thrown away, but the LSB size is smaller. An alternative is to implement linearization with the analog hardware.

A simple approach to a first level linearization of the thermistor output is to use one of the three circuits shown in Figure 5. In Figure 5a, the thermistor is placed in series with a standard resistor (1%, metal film) and a voltage source. The temperature response and linearity of the system shown in Figure 5a. is shown in Figure 6. In this figure, the series thermistor system responds to temperature in a linear manner over a limited temperature range. The linearization resistor's value (R_{SER}) should be equal to magnitude of the thermistor at the mid-point of the temperature range of interest. This creates a response where the output

slope of the resistive network is at its steepest at this mid-point temperature. If high precision is required, this range is typically $\pm 25^\circ\text{C}$ around the nominal temperature of the thermistor at the R_{SER} value.

In Figure 5b., the thermistor is placed in parallel with a standard resistor (R_{PAR}), which creates a composite resistor element. This type of resistive configuration is typically used in system feedback loops and used for automatic gain control circuits.

The resistance to temperature response along with the linearization error of this circuit configuration is shown in Figure 7. Once again, the optimum linearity response of this resistive network is obtained at the point where the thermistor resistance and R_{PAR} are equal.

A third linearization approach is shown in Figure 5c. This circuit combines the parallel configuration in Figure 5b. with an additional reference resistor and a capacitor. The switchable reference is used to charge and discharge the parallel NTC resistance and the reference resistor against the integrating capacitor, C_{INT} . With this circuit, the NTC resistance is biased to a voltage reference and the integrating capacitor charges.

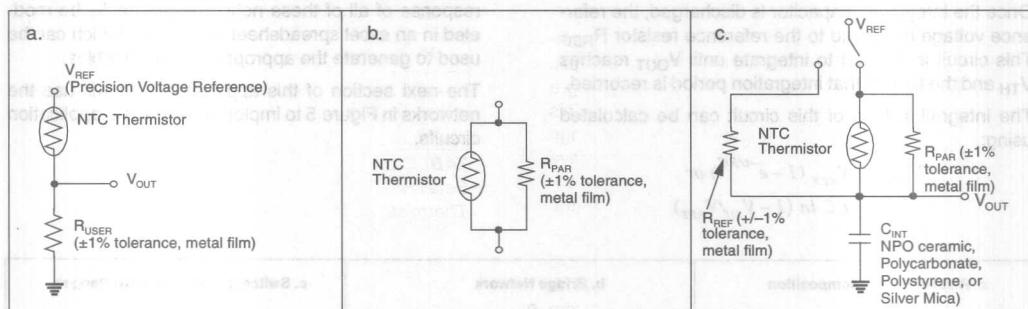


FIGURE 5: The series configuration (a) requires a voltage excitation. The parallel configuration (b) can be used in the feedback loop of an amplifier and does not require a precision source. The parallel configuration can be combined with a capacitor (c) which provides a linear circuit response with time.

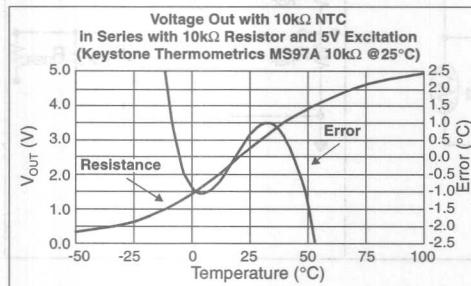


FIGURE 6: The series configuration response of the circuit shown in Figure 5a. has good linear response in a $\pm 25^\circ\text{C}$ range surrounding the temperature where both resistors (NTC and R_{SER}) are equal. The error in this range is typically within $\pm 1\%$. $V_{REF} = 5\text{V}$.

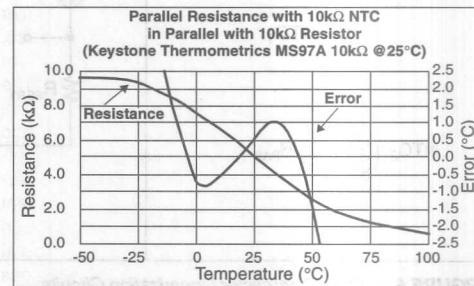
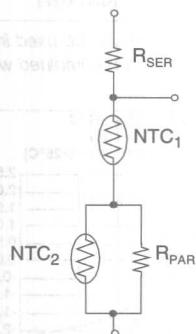


FIGURE 7: The parallel configuration response of the circuit shown in Figure 5c. allows for a counter to be used to determine the relative resistance of the NTC element.

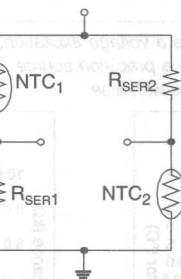
Once the integrating capacitor is discharged, the reference voltage is applied to the reference resistor R_{REF} . This circuit is allowed to integrate until V_{OUT} reaches V_{TH} and the time of that integration period is recorded. The integration time of this circuit can be calculated using:

$$V_{OUT} = V_{REF} (1 - e^{-t/RC}) \text{ or} \\ t = RC \ln (1 - V_{TH}/V_{REF})$$

a. Parallel Series Composition



b. Bridge Network



c. Switchable Temperature Ranges

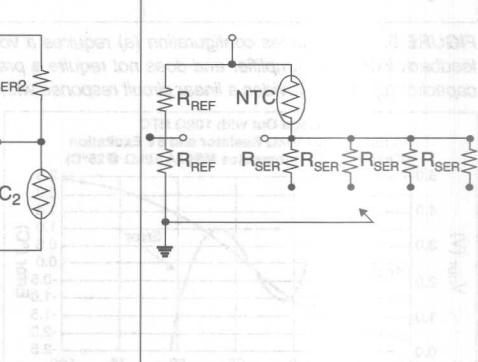


FIGURE 9: Other Thermistor Linearization Circuits.

THERMISTOR SIGNAL CONDITIONING CIRCUITS

There is a large variety of application circuits where the thermistor can be utilized. The three circuits in this application note use the thermistor to implement the cold junction compensation portion of a thermocouple circuit, a linear variable gain versus temperature circuit and an integrated scheme which achieves high accuracy.

Thermocouple Cold Junction Compensation

Although thermocouples can sense temperatures accurately at extreme temperatures or in ambient hostile conditions, a reference temperature is required, if an absolute temperature measurement is desired. (See Microchip's AN684, "Single Supply Temperature Sensing with Thermocouples" for details concerning thermocouple circuit requirements.)

The circuit in Figure 10 is designed to sense the temperature at the isothermal block location with a thermistor. The linearized temperature response of the thermistor is divided down to appropriate levels in order to minimize the EMF voltage errors introduced to the circuit by the parasitic thermocouples on the isothermal block. This style of compensation is done in hardware, requiring no supportive firmware compensation schemes.

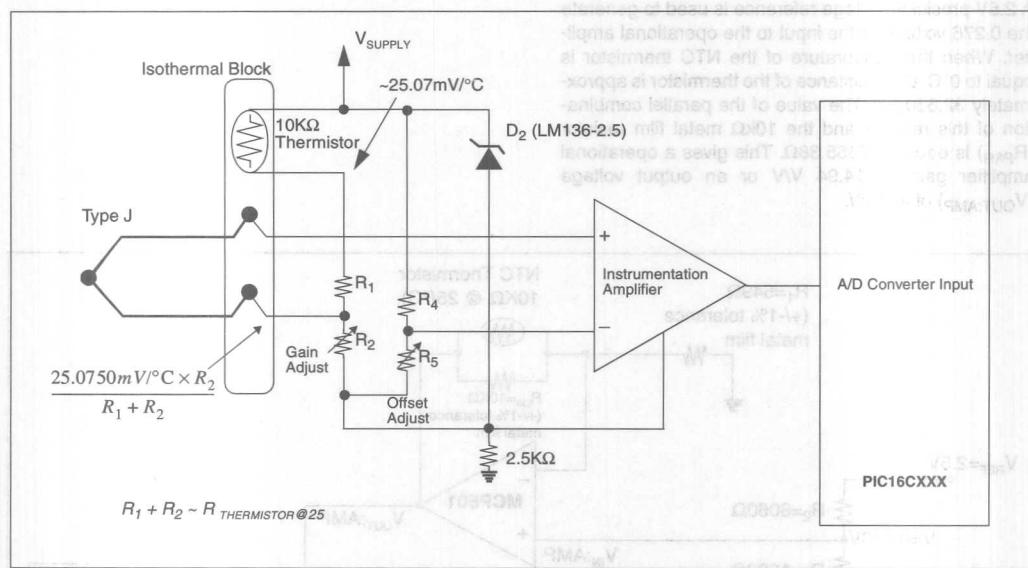


FIGURE 10: A thermistor is used to sense the temperature of the isothermal block in a thermocouple temperature sensing application.

The drift with temperature of the parasitic thermocouples on the isothermal block is approximately $-51\mu\text{V}/^\circ\text{C}$. The thermistor circuit changes by $25.07\text{mV}/^\circ\text{C}$ over the 0 to 50°C linear range given the resistor configuration and the 2.5V excitation voltage. The thermistor drift is divided down using the resistor divider formed with R_1 and R_2 . Appropriate resistor values for R_1 and R_2 with a Type J thermocouple is 100Ω and 49.9kΩ, inclusive. The R_4 and R_5 resistor divider is used to zero offsets in the system as well as implement any required level shifts.

An instrumentation amplifier is used to differentiate the offset error correction circuitry and the Type J thermocouple EMF voltage. (For more details about instru-

mentation amplifiers, see Microchip's AN682, "Using Operational Amplifiers for Analog Gain in Embedded System Design".)

With the thermistor linearization circuitry in place, the voltage changes at the input to the instrumentation amplifier in accordance with temperature changes at the Type J thermocouple measurement site.

The instrumentation amplifier is configured in the appropriate gain for the expected temperature excursions of the Type J thermocouple. The output of the gained analog signal is digitized and used by the microcontroller. With this circuit implementation, the microcontroller is only required to linearize the thermocouple output response.

NOTES:

DS00685B-page 2-126

Temperature Sensing Using an Integrator

The linearization circuit in Figure 5c. is simply implemented with one microcontroller in the signal path as shown in Figure 12.

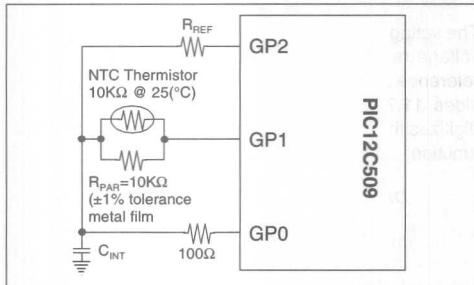


FIGURE 12: This circuit switches the voltage reference on and off at GP1 and GP2. In this manner, the time constant of the NTC Thermistor ($R_{NTC} \parallel R_{PAR}$) and integrating capacitor (C_{INT}) is compared to the time constant of the reference resistor (R_{REF}) and integrating capacitor.

This sensing circuit is implemented by setting GP1 and GP2 of the PIC12C509 as inputs. Additionally, GP0 is set low to discharge the capacitor, C_{INT} . Once C_{INT} is discharged, the configuration of GP0 is changed to an input and GP1 is set to a high output. A timer counts the amount of time before GP0 changes to 1, giving the time, t_1 per Figure 8.

At this point, GP1 and GP2 are again set as inputs and GP0 as an output low. Once the integrating capacitor C_{INT} has time to discharge, GP2 is set to a high output and GP0 as an input. A timer counts the amount of time before GP0 changes to 1, giving the time, t_2 , per Figure 8.

For more details concerning the implementation of this type of integrating circuit, refer to Microchip's AN512, "Implementing Ohmmeter/Temperature Sensor", and AN611, "Resistance and Capacitance Meter Using a PIC16C622".

CONCLUSION

Although the thermistor is non-linear, it can be tamed for a limited temperature range. This allows the design of an inexpensive temperature sensing device which can be used in a variety of Analog-to-Digital Converter applications.

REFERENCES

- Lavenuta, Greg, "Negative Temperature Coefficient Thermistors – the Temp Calibration Standard", *SENSORS*, August, 1997, pg 54.
- Lavenuta, Greg, "Negative Temperature Coefficient Thermistors – Level of Uncertainty", *SENSORS*, June 1997, pg 47.
- Lavenuta, Greg, "Negative Temperature Coefficient Thermistors – Measuring", *SENSORS*, Sept, 1997, pg 48.
- Lavenuta, Greg, "Negative Temperature Coefficient Thermistors" *SENSORS*, May 1997, pg 46.
- Lavenuta, Greg, "Negative Temperature Coefficient Thermistors – Temp Controlled Bath", *SENSORS*, July 1997, pg 17.
- Paillard, Bruno, "Temperature Compensating an Integrated Pressure Sensor", *SENSORS*, Jan. 1998, pg 36.
- Thermometrics Corporation, *Catalog*, 1996.
- Baker, Bonnie, "Temperature Sensing Technologies", AN679, *Microchip Technology Inc.*, 1998.
- "Practical Temperature Measurements", OMEGA CATALOG, pg Z-11.
- Baker, Bonnie, "RTD Temperature Sensing Circuits", AN687, *Microchip Technology Inc.*, 1998.
- Baker, Bonnie, "Single Supply Temperature Sensing with Thermocouples", AN684, *Microchip Technology Inc.*, 1998.
- Baker, Bonnie, "Using Operational Amplifiers for Analog Gain in Embedded System Design", AN682, *Microchip Technology Inc.*, 1998.
- Cox, Doug, "Implementing Ohmmeter/Temperature Sensor", AN512, *Microchip Technology Inc.*
- Richey, Rodger, "Resistance and Capacitance Meter Using a PIC16C622", AN611, *Microchip Technology Inc.*

Brown-Out: A Dirty Little Problem

Brown-out (Figure 2) is a condition where the supply voltage dips or 'sags' down to a safe operating level before returning to a nominal level. This condition can be caused by many different things such as inadequate power regulation, system components turning on or off, system malfunctions, etc. Unfortunately, brown-out conditions often don't show up in the system development stage, but wait until the production run begins with all the system components installed to show their ugly heads. It is often at this point that perplexing prob-

lems are discovered, and eventually tracked down to some kind of brown-out condition. These problems can manifest themselves in many different ways including logic levels being misinterpreted or high current situations by creating invalid CMOS input levels. It is also possible to cause a more insidious problem of corrupting RAM locations inside the microcontroller. This problem can lead to irrational behavior on the part of the microcontroller that does different things at different times and may not show itself at all when an emulator is used to track down the problem.

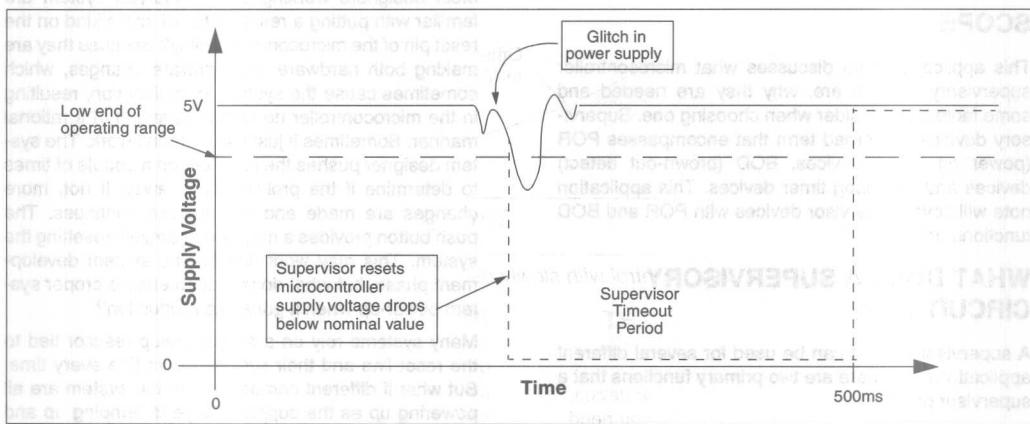


FIGURE 2: Brown-Out Condition
Problems at Power-Down

Most microcontrollers today do not have any on-board POR/BOD protection. Some of them do, but they may not offer adequate protection against some system failures. One system problem that is seen quite frequently is the "Microcontroller running amok" problem that occurs when the supply voltage is ramped down very slowly, such as when a bench power supply is turned down manually or during the decay of a battery supply. When this situation occurs, it is possible for many microcontrollers to begin running through its code in a somewhat random manner. There may not be enough voltage to sustain RAM locations, so the program counter as well as any other variable stored in RAM may not contain valid data. This provides the means for the micro to execute any or all portions of the code stored in program memory with indeterminate values in all RAM locations.

Obviously, the longer it takes for the supply to ramp down the greater the danger of this situation occurring and causing problems. See Figure 3. For some systems, this situation may not cause any problems more serious than some spurious data sent to a display as the system is powered down. However, if the system contains other components that work to a lower voltage such as EEPROM devices, the problem becomes potentially more serious. EEPROM devices are avail-

able on the market that work down to 1.8V and may respond to commands as low as 1.2V. If the microcontroller executes a portion of its code that controls writing to the EEPROM, then there is the distinct possibility that random data will be written to the EEPROM device, which may or may not be discovered when the system is powered up the next time. This problem very often does not show up in the system development phase because the system is not being powered up and down on a regular basis, or it is powered from a supply different from the one used in production. It often shows up when the system goes into production and the system is being tested at different stages of the production line with different power supplies. A typical scenario: Data is written into the EEPROM and the system is tested as good and then powered down. At the next station it is discovered that the EEPROM data has been corrupted. This often results in a call to the EEPROM vendor with complaints of data retention problems, when the actual problem was the microcontroller sending write commands to the EEPROM during power down.



Understanding and Using Supervisory Circuits

Author: Bruce Negley
Microchip Technology Inc.

SCOPE

This application note discusses what microcontroller supervisory devices are, why they are needed and some factors to consider when choosing one. Supervisory devices is a broad term that encompasses POR (power on reset) devices, BOD (brown-out detect) devices and watchdog timer devices. This application note will cover supervisor devices with POR and BOD functions only.

WHAT DOES A SUPERVISORY CIRCUIT DO?

A supervisory circuit can be used for several different applications, but there are two primary functions that a supervisor provides:

1. During a power up sequence, the device holds a microcontroller in reset until the system power has come up to the correct level and stabilized (the POR function), and
2. reset the controller immediately if the power drops below a nominal value either at power down or during a 'brown-out' condition.

Some supervisor devices also provide things like low battery warning, watchdog timer and other more elaborate functions that are beyond the scope of this application note.

WHY DO I NEED A SUPERVISORY CIRCUIT ANYWAY?

One question system designers may ask themselves is, "Why do I need one of these things anyway?" There are 3 situations that you must consider when answering this question:

1. What would happen to the microcontroller (or other devices in the system) if there was noise on the supply voltage as it powers up?
2. What would happen if there is a glitch on the power supply while the system is running?
3. What does the microcontroller do when the system power is turned off?

If you ponder these questions and have visions of phone calls from angry customers, then you might consider using a supervisor device.

IN THE BEGINNING: POWER-UP PROBLEMS

Most designers working on a prototype system are familiar with putting a reset switch of some kind on the reset pin of the microcontroller. Why? Because they are making both hardware and firmware changes, which sometimes cause the system to malfunction, resulting in the microcontroller no longer behaving in a rational manner. Sometimes it just plain doesn't work. The system designer pushes the reset button a couple of times to determine if the problem goes away. If not, more changes are made and the process continues. The push button provides a means of manually resetting the system. This may work fine for the system development phase, but what do you do to ensure proper system power-up when it goes into production?

Many systems rely on a simple pullup resistor tied to the reset line and their system works fine every time. But what if different components in the system are all powering up as the supply voltage is ramping up and noise is injected onto the supply line? Most microcontrollers have specs that describe power up ramps for proper initialization of the controller. A glitch on the supply line may very well cause the microcontroller (or some other component) to power-up incorrectly and prevent the system from operating as intended. See Figure 1. A supervisor device solves this problem by holding the microcontroller in reset until the power has reached a stable level. Timeout periods vary for different devices but typical values are 150ms - 500ms. When the timeout period is complete, the device will release the reset line and allow the microcontroller to begin execution of its code.

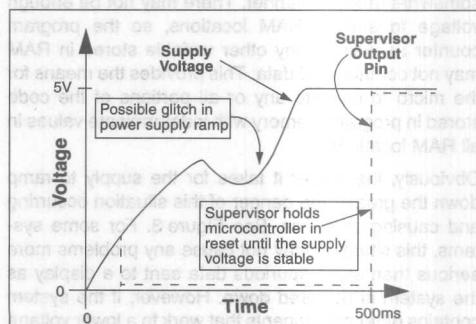
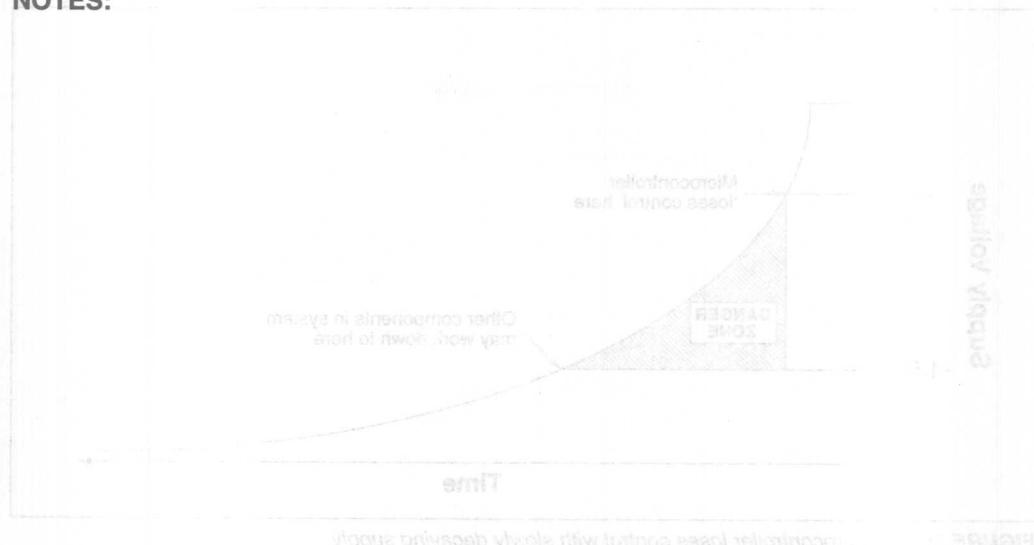


FIGURE 1: POR Function

NOTES:



Many microcontrollers have a wide power range for their devices. This is often due to the fact that the device is designed to work with a wide range of voltages. For example, some microcontrollers can operate from 3.0V to 5.5V. This allows them to be used in a variety of applications, such as mobile phones, PDAs, and laptop computers. However, it is important to note that the power consumption of a microcontroller increases as its operating voltage increases. This is because the current required to drive the internal logic and memory cells increases with the voltage.

Some microcontrollers have a second power source, which is often used for the external memory. This allows the microcontroller to operate with two different voltages. For example, the MCP1601 has two power inputs: VDD and VSS. The VDD input is used for the internal logic, while the VSS input is used for the external memory.

CONCLUSIONS

Using a separate power source for the external memory can help to reduce power consumption. This is because the current required to drive the external memory is lower than the current required to drive the internal logic. In addition, using a separate power source for the external memory can help to reduce noise levels. This is because the noise generated by the external memory is lower than the noise generated by the internal logic.

SO HOW TO CHOOSE THE RIGHT DEVICE

For those who are new to microcontroller design, choosing the right device can be a daunting task. There are many factors to consider, including the type of application, the cost of the device, and the availability of support. One way to approach this is to start by defining the requirements of the application. You will then need to choose a device that can meet these requirements. You will also need to consider the cost of the device. It is important to remember that the cost of a microcontroller is not the only factor to consider. The performance, reliability, and ease of use are also important factors.

Microcontroller Power (V)	Optimal Power (V)	Minimum Power (V)
3.0	3.0	3.0
3.3	3.3	3.3
3.6	3.6	3.6
3.9	3.9	3.9
4.2	4.2	4.2
4.5	4.5	4.5

TABLE 2. Optimal Power Range

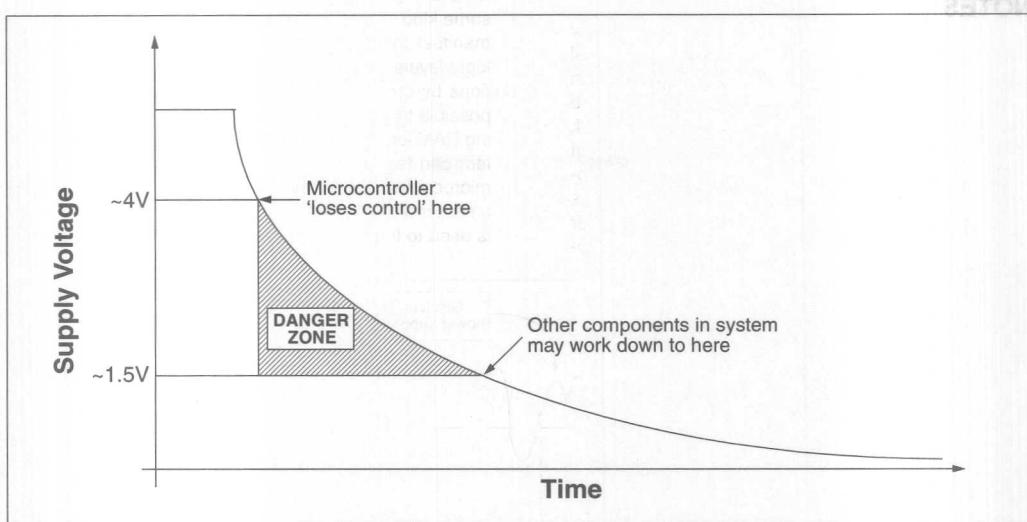


FIGURE 3: Microcontroller loses control with slowly decaying supply

SO HOW DO I CHOOSE THE RIGHT DEVICE?

For the standard POR/BOD type of supervisor device, there are really only a couple of factors that you need to consider when making your choice. The major factors to consider are: reset voltage, output driver type, and reset polarity. Most supervisor devices come in a variety of reset voltages to support both 5V and 3V systems. Table 1, below shows typical reset voltage ranges. Choosing the correct trip point depends mainly on the operating range of the controller you are using and the variation of your supply voltage. You want to choose the highest trip point you can that will not interfere with the normal variations of your supply voltage. For a typical microcontroller, it might operate at 5V $\pm 10\%$ or 4.5V - 5.5V. Choosing a device with a trip point range of 4.5V - 4.75V will ensure that the controller is reset before the low end of the operating range is reached.

Minimum Trip Point (V)	Typical Trip Point (V)	Maximum Trip Point (V)
2.55	2.625	2.7
2.85	2.925	3.0
3.0	3.075	3.15
4.25	4.375	4.50
4.35	4.475	4.60
4.50	4.625	4.75
4.60	4.725	4.85

TABLE 1: Typical Trip Point Values

Many vendors also provide different output driver options for their devices. The usual choices are open drain, open drain with internal pull-up and standard push-pull output drivers. The open drain options allow more than one source to pull the reset line to the reset state, such as a pushbutton or some other component that has the ability to reset the controller such as an over-temperature safety switch.

Since some microcontrollers have a low active reset line and some are high active, you must also choose a reset device with the correct polarity. For reference, the MCP100/120/130 are all active low devices and the MCP101 is active high.

CONCLUSIONS

Using supervisory circuits can protect microcontroller based systems from a number of power-related problems. If you are experiencing problems in your system that are not making sense, it may be power related and if so, it may be beneficial to add a supervisory device to the system. This application note provides some guidelines that you can use in determining what the problem might be and what device should be chosen to solve the problem.



MICROCHIP

AN687

Precision Temperature Sensing with RTD Circuits

Author: **Bonnie C. Baker**
Microchip Technology Inc.

INTRODUCTION

One of the most widely measured phenomena in the process control environment is temperature. Common elements such as Resistance Temperature Detectors (RTDs), thermistors, thermocouples or diodes are used to sense absolute temperatures as well as changes in temperature. For an overview and comparison of these sensors, refer to Microchip's AN679, "Temperature Sensing Technologies".

Of these technologies, the platinum RTD temperature sensing element is the most accurate and stable over time and temperature. RTD element technologies are constantly improving, further enhancing the quality of the temperature measurement (see Figure 1). Typically, a data acquisition system conditions the analog signal from the RTD sensor, making the analog translation of the temperature usable in the digital domain.

This application note focuses on circuit solutions that use Platinum RTDs in the design. Initially, the RTD temperature sensing element will be compared to the negative temperature coefficient (NTC) thermistor, which is also a resistive temperature sensing element. In this forum the linearity of the RTD will be presented along with calibration formulas that can be used to improve the off the shelf linearity of the element. If more information is needed concerning the thermistor temperature sensor, refer to Microchip's AN685, "Thermistors in Single Supply Temperature Sensing Circuits". Finally, the signal conditioning path for the RTD system will be covered with complete application circuits from sensor or microprocessor.

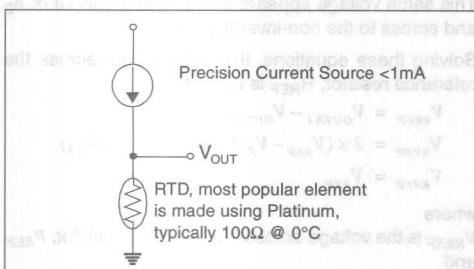


FIGURE 1: Unlike thermistors, RTD temperature sensing elements require current excitation.

RTD OVERVIEW

The acronym "RTD" is derived from the term "Resistance Temperature Detector". The most stable, linear and repeatable RTD is made of platinum metal. The temperature coefficient of the RTD element is positive. This is in contrast to the NTC thermistor that has a negative temperature coefficient as shown graphically in Figure 2. An approximation of the platinum RTD resistance change over temperature can be calculated by using the constant $0.00385\Omega/\Omega^{\circ}\text{C}$. This constant is easily used to calculate the absolute resistance of the RTD at temperature.

$$RTD(T) = RTD_0 + T \times RTD_0 \times 0.00385\Omega/\Omega^{\circ}\text{C}$$

where

$RTD(T)$ is the resistance value of the RTD element at temperature (Celsius),

RTD_0 is the specified resistance of the RTD element at 0°C , and

T is the temperature environment that the RTD is placed (Celsius).

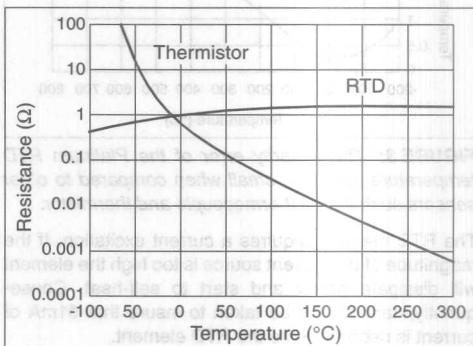


FIGURE 2: The temperature versus resistance characteristics of the RTD sensing element is considerably different than the thermistor sensor element. The RTD sensing element has a positive temperature coefficient and is considerably more linear.

The RTD element resistance is extremely low as compared to the resistance of an NTC thermistor element which ranges up to $1\text{M}\Omega$ at 25°C . Typical specified 0°C values for RTDs are 50, 100, 200, 500, 1000 or 2000Ω . Of these options, the 100Ω platinum RTD is the most stable over time and linear over temperature.

2

Application Notes

AN687

CONCLUSION

Although the RTD requires a more circuitry in the signal conditioning path than the thermistor or the silicon temperature sensor, it ultimately provides a high precision, relatively linear result over a wider temperature range. If further linearization is performed in the processor, the RTD circuit can achieve $\pm 0.01^\circ\text{C}$ accuracy.

REFERENCES

Baker, Bonnie, "Temperature Sensing Technologies", AN679, Microchip Technology Inc.

"Practical Temperature Measurements", OMEGA CATALOG, pg Z-11 (concrete to east) (leads east to)

Baker, Bonnie, "Single Supply Temperature Sensing with Thermocouples", AN684, Microchip Technology Inc.

Baker, Bonnie, "Using Operational Amplifiers for Analog Gain in Embedded System Design", AN682, Microchip Technology Inc.

Baker, Bonnie, "Thermistors in Single Supply Temperature Sensing Circuits", AN685, Microchip Technology Inc.

Hyde, Darrell, "Evaluating Thin Film RTD Stability",
SENSORS, OCT. 1997, pg 79

Madden, J.R., "Refresher on Resistance Temperature Devices", *SENSORS*, Sept., 1997, pg 66

Li, Xumo, "Producing Higher Accuracy From SPRTs (Standard Platinum Resistance Thermometer)", *MEASUREMENT & CONTROL*, June, 1996, pg118

This circuit generates a current source that is ratiometric to the voltage reference. The same voltage reference can be used in other portions of the circuit, such as the analog-to-digital (A/D) converter reference.

Absolute errors in the circuit will occur as a consequence of the absolute voltage of the reference, the initial offset voltages of the operational amplifiers, the output swing of A_1 , mismatches between the resistors, the absolute resistance value of R_{REF} and the RTD element. Errors due to temperature changes in the circuit will occur as a consequence of the temperature drift of the same elements listed above. The primary error sources over temperature are the voltage reference, offset drift of the operational amplifiers and the RTD element.

RTD SIGNAL CONDITIONING PATH

Changes in resistance of the RTD element over temperature is usually digitized through an A/D conversion as shown in Figure 5. The current excitation circuit shown in Figure 4 is used to excite the RTD element. With this style of excitation, the magnitude of the current source can be tuned to 1mA or less by adjusting R_{REF} . The voltage drop across the RTD element is sensed by A_3 then gained and filtered by A_4 . With this circuit, a three-wire RTD element is selected. This configuration minimizes errors due to wire resistance and wire resistance drift over temperature.

In this circuit, the RTD element equals 100Ω at 0°C . If the RTD is used to sense temperature over its entire range of -200 to 600°C , the range of resistance produced by the RTD would be nominally 23Ω to 331Ω .

Since the resistance range is relatively low, wire resistance and wire resistance change over temperature can skew the measurement of the RTD element. Consequently, a three-wire RTD device is used to reduce these errors.

The errors contributed by the wire resistances, R_{W1} and R_{W3} , is subtracted from the circuit with the A_3 the operational amplifier circuit. In this configuration, R_1 and R_2 are equal and relatively high. The value of R_3 is selected to insure that the leakage currents through the resistor does not introduce errors to the current to the RTD element. The transfer function of this portion of the circuit is:

$$V_{OUTA3} = (V_{IN} - V_{w1})(1 + R_2/R_1) - V_{IN}(R_2/R_1)$$

where

$$V_{IN} = V_{W1} + V_{RTD} + V_{W3},$$

V_{Wx} is the voltage drop across the wires to and from the RTD and

V_{OUTA3} is the voltage at the output of A_3 .

If it is assumed that

$$R_1 = R_2 \text{ and } R_{W1} = R_{W3}$$

the transfer function above reduces to:

$$V_{OUTA3} = V_{RTD}$$

The voltage signal at the output of A_3 is filtered with a 2nd order, low pass filter created with A_4 , R_3 , C_3 , R_4 , and C_4 . This same signal is also gained by the resistors R_5 and R_6 .

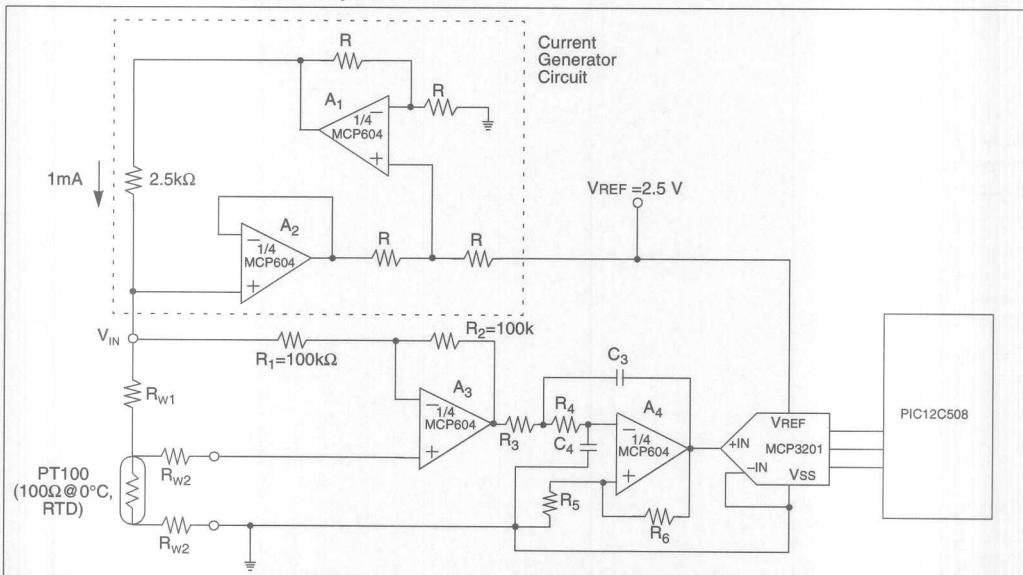


FIGURE 5: This circuit uses an RTD temperature sensitive element to measure temperatures from -200 to 600°C . The current generator circuit from Figure 4 excites the sensor. An operational amplifier, A_3 , is used to zero wire resistance error. A fourth amplifier, A_4 , is used to gain the signal and filter possible alias interference. A 12-bit converter, MCP3201, converts the voltage across the RTD to digital code for the 8-pin controller, PIC12C508.

ONE MAJOR STEP TOWARDS DISASTER

The size of this circuit seems manageable. So small that one may be tempted to use an auto router layout tool. If this type of tool is used, it should be used carefully. If the tool is capable of implementing restrictions into the layout implementation, the layout design may have a fighting chance. If restrictions are not implemented by the auto routing tool, the best approach is to not use it at all.

GENERAL LAYOUT GUIDELINES

Device Placement

Device placement is critical. In general, there are some noise sensitive devices in this layout and other devices that are major problem creators. Here is a quick way to identify the good, from the bad, from the ugly.

1. Separate the circuit devices into two categories: high speed (>40MHz) and low speed.
2. Separate the above categories into three sub-categories: pure digital, pure analog, and mixed signal.

The board layout strategy should map the diagram shown in Figure 2. Notice the relationship of digital versus analog and high speed versus slower speeds to the board connector.

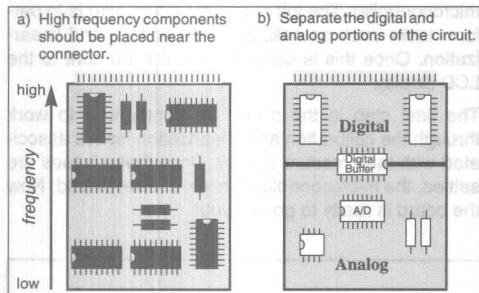


FIGURE 2: The placement of an active component on a PCB is critical in precision 12-bit+ circuits.

In Figure 2b, the digital and analog circuit is shown as being separate from the digital devices, which are closest to the connector or power supply.

The pure analog devices are furthest away for the digital devices to insure that switching noise is not coupled into the analog signal path.

The treatment of the A/D Converter in layout varies from technology to technology. For instance, if the A/D Converter uses a Successive Approximation Register (SAR) design approach, the entire device should be connected to the analog power and ground planes. A common error is to have the converter straddle the analog and digital planes. This strategy may work, but as the accuracy specifications of the A/D Converter improve the digital ground and power plane noise begins to cause problems. For high resolution SAR converters, a digital buffer should be used to isolate the converter from bus activity on the digital side of the circuit.

In contrast, if the A/D Converter is designed using a delta-sigma technology, it should straddle the analog and digital planes. This is due to the fact that the Delta-Sigma Converter is primarily a digital IC.

Ground and Power Supply Strategy

Once the general vicinity of the devices are determined, the ground planes and power planes should be defined. The strategy of the implementation of these planes are a bit tricky.

First of all, assuming that a ground plane is not needed is a dangerous assumption in any circuit with analog and/or mixed signal devices. Ground noise problems are more difficult to deal with than power supply noise problems because analog signals are most typically referenced to ground. For instance, in the circuit shown in Figure 1, the A/D Converter's inverting input pin (MCP3201) is connected to ground. Additionally, the negative side of the pressure sensor is also connected to ground.

A layout for the circuit in Figure 1 is shown in Figure 3. This layout implementation does not have a ground or power plane on the board.

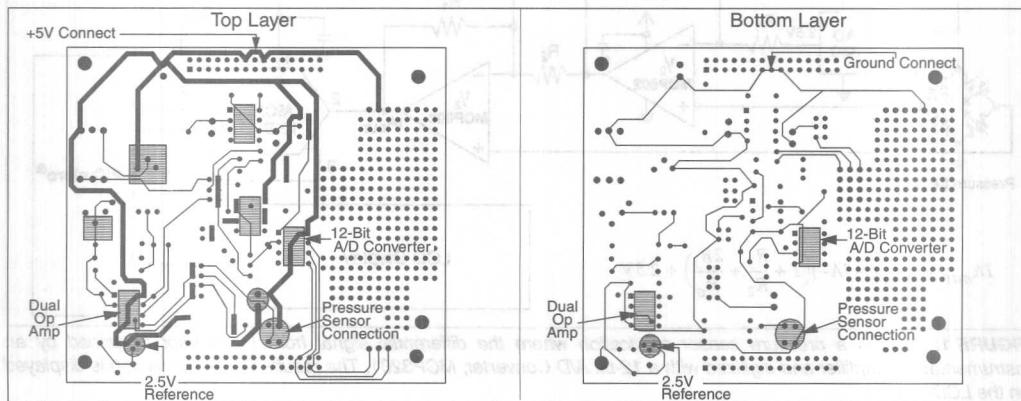


FIGURE 3: Layout of the top and bottom layers of the circuit in Figure 1. Note that this layout does not have a ground or power plane.



MICROCHIP

AN688

Layout Tips for 12-Bit A/D Converter Application

Author: Bonnie C. Baker

Microchip Technology Inc.

INTRODUCTION

This Application Note originally started as a "cook book" for a true 12-bit layout. The assumption of this type of approach is that a reference design could be provided, which easily could be used for every layout implementation. But, the notion of this approach is fairly unrealistic. There are a multitude of successful ways to layout out systems with 12-bit Analog-to-Digital (A/D) Converters and each layout is highly dependent on the number of devices in the circuit, the types of the devices (digital or analog) and the environment that the final product will reside in. Given all of these variables, it could easily be demonstrated that one successful layout out that provides twelve noise free bits from an analog signal may easily fail in another setting.

Because of the complexity of this problem, this Application Note will provide basic guidelines, ending with a review of issues to be aware of. Throughout the application note, examples of good layout and bad layout implementations will be presented. This will be done in the spirit of discussing concepts and not with the intent of recommending one layout as the only one to use.

GETTING A GOOD START

Imagine that the task at hand is to design a pressure sensing circuit that will accurately measure the pressure and present the results on an LCD display screen. Seems easy enough.

The circuit diagram for this system is shown in Figure 1. The pressure sensor that is chosen for the job is a piezo resistive sensor that is configured as a four element bridge. The particular sensor that is selected requires voltage excitation. The full swing output of the sensor is a small (10s of millivolts) differential signal that most appropriately is gained by an operational amplifier structure that also converts the differential output of the sensor to a single ended analog signal. A 12-bit converter is chosen to match the precision of the pressure sensor. Once the converter digitizes the voltage presented at its input, the digital code is sent to a microcontroller. The job of the microcontroller is to perform tasks such as calibration corrections and linearization. Once this is done, the results are sent to the LCD display.

The final step in the circuit development is to work through the calibration and linearization issues associated with the pressure sensor. Once these issues are settled, the microcontroller firmware is developed. Now the board is ready to go to layout.

2

Application
Notes

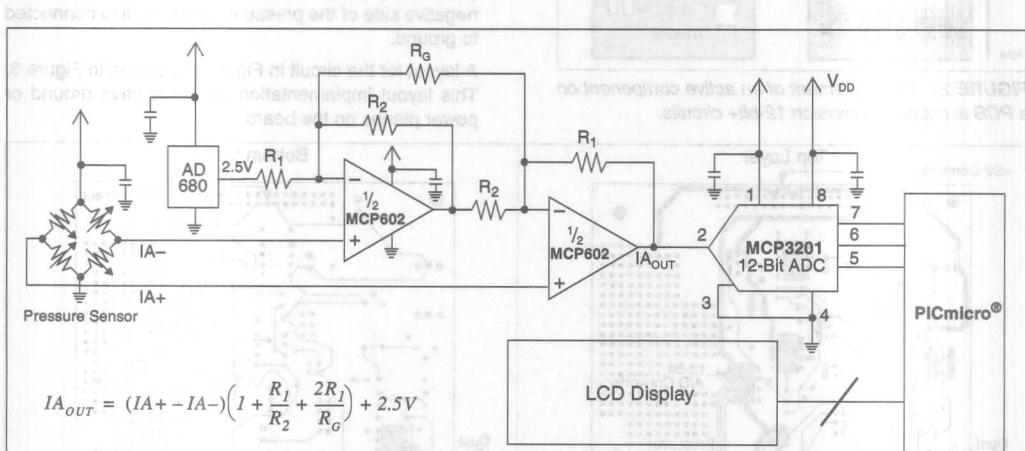


FIGURE 1: This is a pressure sensor application where the differential signal from the sensor is gained by an instrumentation amplifier and digitized with a 12-bit A/D Converter, MCP3201. The results of the conversion is displayed on the LCD display.

Generally speaking, the signal traces on the board (both digital and analog) should be as short as possible. This basic guideline will minimize the opportunities for extraneous signals to couple into the signal path.

One area to be particularly cautious of is the input terminals of analog devices. These input terminals normally have a higher impedance than the output or power supply pins. As an example, the voltage reference input pin to the analog to digital converter is most sensitive while a conversion is occurring. With the type of 12-bit converter shown in Figure 1, the input terminals (IN+ and IN-) are also sensitive to injected noise.

Another potential for noise injection into the signal path is the input terminals of an operational amplifier. These terminals have typically 10^9 to 10^{13} Ω input impedance.

These high impedance input terminals are sensitive to injected currents. This can occur if the trace from a high impedance input is next to a trace that has fast changing voltages, such as a digital or clock signal. When a high impedance trace is in close proximity to a trace with these types of voltage changes, charge is capacitively coupled into the high impedance trace.

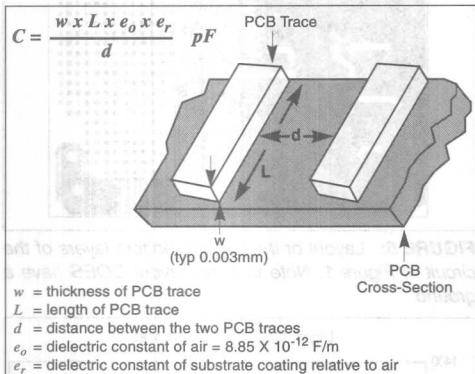


FIGURE 7: A capacitor can be constructed on a PCB by placing two traces in close proximity. With this PCB capacitor, signals can be coupled between the traces.

As shown in Figure 7, the value of the capacitance between two traces is primarily dependent on the distance (d) between the traces and the distance that the two traces are in parallel (L). From this model, the amount of current generated into the high impedance trace is equal to:

$$I = C \delta V / \delta t$$

where

I equals the current that appears on the high impedance trace

C equals the value of capacitance between the two PCB traces

δV equals the change in voltage of the trace that is switching, and

δt equals the amount of time that the voltage change took to get from one level to the next.

A good rule concerning by-pass capacitors is to always include them in the circuit. If they are not included, the power supply noise may very well eliminate any chance for 12-bit precision.

By-pass capacitors belong in two locations on the board: one at the power supply ($10\mu\text{F}$ to $100\mu\text{F}$ or both) and one for every active device (digital and analog). The value of the device's by-pass capacitor is dependent on the device in question. If the bandwidth of the device is less than or equal to ~1MHz, a $1\mu\text{F}$ will reduce injected noise dramatically. If the bandwidth of the device is above ~10MHz, a $0.1\mu\text{F}$ capacitor is probably appropriate. In between these two frequencies, both or either one could be used. Refer to the manufacturer's guidelines for specifics.

Every active device on the board requires a by-pass capacitor. The by-pass capacitor must be placed as close as possible to the power supply pin of the device as shown in Figure 5. If two by-pass capacitors are used for one device, the smaller one should be closest to the device pin. Finally, the lead length of the by-pass capacitor should be as short as possible.

To illustrate the benefits of by-pass capacitors, data is collected from the layout shown in Figure 5, minus the by-pass capacitors. This data is shown in Figure 8.

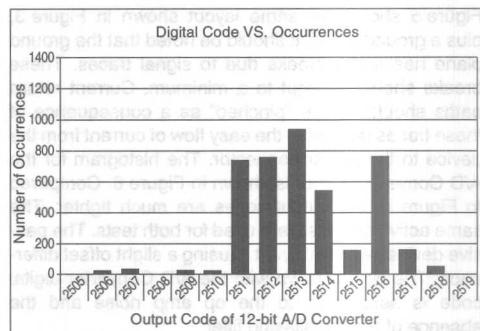


FIGURE 8: This is a histogram of 4096 samples from the output of the A/D Converter on the PCB that has a ground plane as shown in the PCB layout in Figure 3. With this circuit implementation, all by-pass capacitors have been removed.

The histogram shows the digital code versus the number of occurrences. The distribution is highly skewed, with the most frequent codes (M11, M12, M13) having approximately 800 occurrences each, and others being much lower. This indicates that without by-pass capacitors, the power supply noise is significant enough to affect the digital output codes.

With this circuit layout, the controller is dedicated to interfacing with the converter and sending the converter's results to the LCD display. The digital output of the converter over time is shown in Figure 4. This data was collected with no excitation being applied to the sensor.

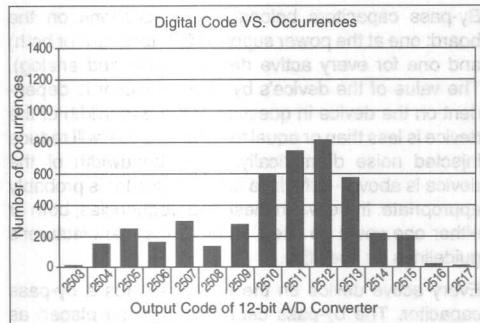


FIGURE 4: This is a histogram of 4096 samples from the output of the A/D Converter from a PCB that does not have a ground or power plane as shown in the PCB layout in Figure 3. The by-pass capacitors are installed.

When determining the grounding strategy of a board, the task at hand should actually be to determine if the circuit can work adequately with just one ground plane or does it need multiple planes.

Figure 5 shows the same layout shown in Figure 3, plus a ground plane. It should be noted that the ground plane has a few breaks due to signal traces. These breaks should be kept to a minimum. Current return paths should not be "pinched" as a consequence of these traces restricting the easy flow of current from the device to the power connector. The histogram for the A/D Converter output is shown in Figure 6. Compared to Figure 4, the output codes are much tighter. The same active devices were used for both tests. The passive devices were different causing a slight offset difference. The noise shown with the A/D Converter digital code is assignable to the op amp noise and the absence of an anti-aliasing filter.

If the circuit has a "minimum" amount of digital circuitry on board, a single ground plane and a single power plane may be appropriate. The qualifier "minimum" is defined by the board designer. The danger of connecting the digital and analog ground planes together is that the analog circuitry can pick-up the noise on the supply pins and couple it into the signal path. In either case, the analog and digital grounds and power supplies should be connected together at one or more points in the circuit to insure that the power supply, input and output ratings of all of the devices are not violated.

The inclusion of a power plane in a 12-bit system is not as critical as the required ground plane. Although a power plane can solve many problems, power noise can be reduced by making the power traces two or three times wider than other traces on the board and by using by-pass capacitors effectively.

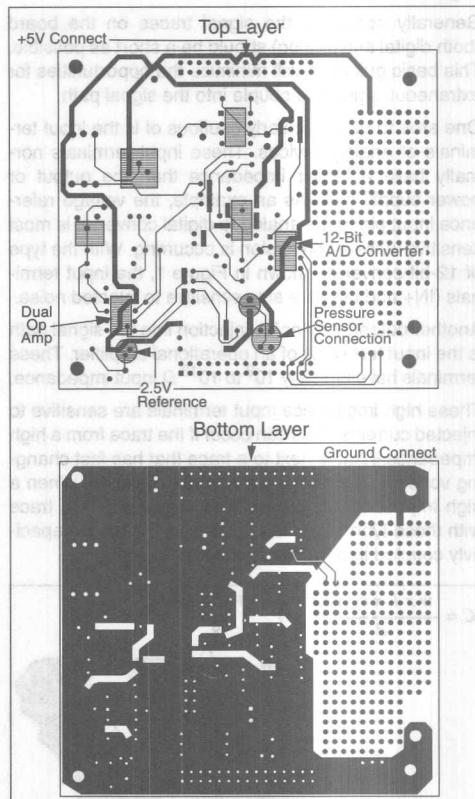


FIGURE 5: Layout of the top and bottom layers of the circuit in Figure 1. Note that this layout DOES have a ground.

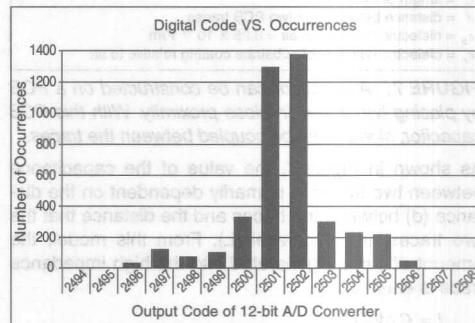


FIGURE 6: This is a histogram of 4096 samples from the output of the A/D Converter on the PCB that has a ground plane as shown in the PCB layout in Figure 5. Note that the power traces are made considerably wider than the signal traces in order to reduce power supply trace inductance. This circuit has all by-pass capacitors installed.

AN688

NOTES:

PCB DESIGN CHECK LIST

Good 12-bit layout techniques are not difficult to master as long as a few guidelines are considered.

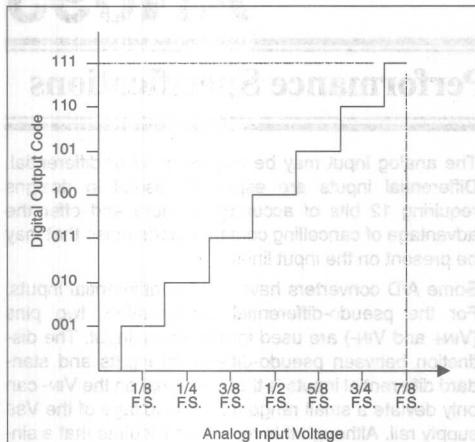
1. Check device placement versus connectors. Make sure that high speed devices and digital devices are closest to the connector.
2. Always have at least one ground plane in the circuit.
3. Make power traces wider than other traces on the board.
4. Review current return paths and look for possible noise sources on ground connects. This is done by determining the current density at all points of the ground plane and the amount of possible noise present.
5. By-pass all devices properly. Place the capacitors as close to the power pins of the device as possible.
6. Keep all traces as short as possible.
7. Follow all high impedance traces looking for possible capacitive coupling problems from trace to trace.

REFERENCES

Morrison, Ralph, "Noise and Other Interfering Signals", *John Wiley & Sons, Inc.*, 1992

Baker, Bonnie, "Noise Sources in Applications Using Capacitive Coupled Isolated Amplifiers", *AB-047, Burr-Brown Corporation*

FIGURE 2: IDEAL A/D TRANSFER FUNCTION



Code Width

The width of a given output code is the range of analog input voltages for which that code is produced. The code widths are referenced to the weight of 1 least significant bit (LSb), which is defined by the resolution of the converter and the analog reference voltage. So 1 LSb = $V_{REF}/2^N$, where N is the number of bits of resolution. For example, if a 4.096 volt reference is used with a 12-bit converter, 1 LSb will have a weight of 4.096 V/ 2^{12} , or 1 mV. All codes will have a width of 1 LSb for an ideal A/D converter.

Resolution and Accuracy

Resolution and accuracy are terms that are often interchanged when the performance of an A/D converter is discussed. The resolution of an A/D converter is specified in bits and determines how many distinct output codes (2^N) the converter is capable of producing. For example, an 8-bit A/D converter produces 2^8 , or 256, output codes.

The accuracy of the A/D converter determines how close the actual digital output is to the theoretically expected digital output for a given analog input. In other words, the accuracy of the converter determines how many bits in the digital output code represent useful information about the input signal. The accuracy of the A/D converter is a function of its internal circuitry and noise from external sources connected to the A/D input.

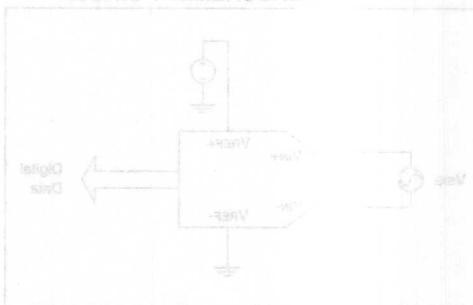
In some cases, extra bits of resolution that are beyond the accuracy of the A/D converter can be beneficial. Delta-Sigma A/D converters, for example, can provide resolutions as high as 24 bits. A given 24-bit Delta-Sigma converter may only provide 16 bits of accuracy. In this case, the 8 LSb's represent random noise produced in the converter. However, these noise

bits are used with digital filter algorithms to increase the useful measurement resolution at the expense of a lower sampling bandwidth.

Acquisition Time

A successive approximation (SAR) A/D converter will have a track and hold circuit at the analog input. Internally, the track and hold circuit is implemented as a charge holding capacitor that is disconnected from the analog input pin just before the A/D conversion begins. The holding capacitor must be given sufficient time to charge to its final value, or errors will be introduced into the conversion. The acquisition time that must be allowed is a function of the holding capacitor value, source impedance and internal resistances associated with the input circuit. Figure 3 shows a typical model for the analog input of a SAR A/D converter. The input model parameters will vary, so the designer should refer to the device data sheet to ensure that the proper acquisition time is provided based on the input circuit that is used in the design.

THE IDEAL A/D CONVERTER



THE IDEAL D/A CONVERTER

The ideal D/A converter is a digital-to-analog converter that produces a continuous analog output voltage. It is characterized by its high resolution and low noise. The output voltage is determined by the digital input code and the reference voltage.

The graph illustrates the ideal transfer function for a D/A converter. The vertical axis represents the Output Voltage, and the horizontal axis represents the Digital Input. The transfer function is a piecewise constant function that maps the digital input code into an analog output voltage. The steps occur at integer digital input values.

Digital Input	Output Voltage
0	0
1	V _{REF} /2
2	2V _{REF} /2
3	3V _{REF} /2
4	4V _{REF} /2
5	5V _{REF} /2
6	6V _{REF} /2
7	7V _{REF} /2



Understanding A/D Converter Performance Specifications

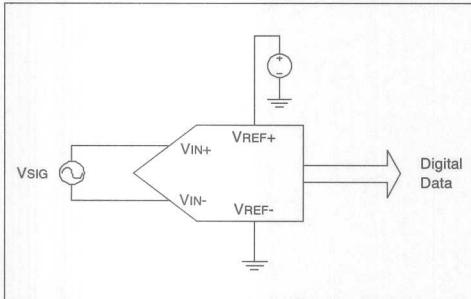
Authors: Steve Bowling
Microchip Technology Inc.

INTRODUCTION

The purpose of this application note is to describe the specifications used to quantify the performance of A/D converters and give the reader a better understanding of the significance of those specifications in an application. Although the information presented here is applicable to all A/D converters, specific attention is given to features of the stand-alone and PICmicro® A/D converters produced by Microchip Technology.

Figure 1 shows a block diagram of the typical A/D converter measurement circuit.

FIGURE 1: BASIC A/D CONVERTER MEASUREMENT CIRCUIT



THE IDEAL A/D CONVERTER

The ideal A/D converter produces a digital output code that is a function of the analog input voltage and the voltage reference input. The formula for the A/D converter digital output is given by Equation 1.

EQUATION 1: A/D OUTPUT

$$\text{OutputCode} = F.S. \times \frac{V_{IN+} - V_{IN-}}{V_{REF+} - V_{REF-}} = F.S. \times \frac{V_{IN}}{V_{REF}}$$

The analog input may be single-ended or differential. Differential inputs are especially useful in designs requiring 12 bits of accuracy or more and offer the advantage of cancelling common mode noise that may be present on the input lines.

Some A/D converters have pseudo-differential inputs. For the pseudo-differential configuration, two pins (V_{IN+} and V_{IN-}) are used for the signal input. The distinction between pseudo-differential inputs and standard differential inputs is that the signal on the V_{IN-} can only deviate a small range from the voltage of the V_{SS} supply rail. Although this restriction requires that a single-ended source is connected to the A/D converter, the input stage maintains the ability to cancel small common-mode fluctuations on the input pins.

The voltage reference for the A/D converter may be provided internally or by an external source. Since the accuracy of the measurement results is directly affected by the reference, it is important that the reference source be stable over time and temperature. For low cost converters, the reference input is often implemented as a single-ended input. In this case, one pin is used for the reference input and the input voltage range for the converter is determined by V_{SS} and V_{REF} . For higher performance converters, two voltage reference pins are typically provided. The input voltage range for these converters is determined by the voltage difference between V_{REF+} and V_{REF-} . In either case, the voltage range for the reference inputs is usually restricted by the V_{DD} and V_{SS} power supply rails.

Although a "real world" A/D converter will have higher resolution, a theoretical 3-bit A/D converter will be used here to demonstrate the performance of the ideal converter and the various sources of error. Figure 2 shows the transfer function of the ideal 3-bit A/D converter. As the transfer function indicates, the ideal 3-bit A/D converter provides eight equally spaced digital output codes over the analog input voltage range.

Each digital output code represents a fractional value of the reference voltage. The largest value that can be obtained from the A/D converter is $(N-1)/N$, where N is the resolution in bits. Referring to Figure 2, the largest output value that the 3-bit A/D converter can produce is $7/8^{\text{ths}}$ of the full-scale reference voltage.

Temperature Dependent Reference

A temperature dependent reference voltage can be constructed using thermistor/resistive parallel combination illustrated in Figure 5b, as the feedback element in an operational amplifier circuit. The implementation of this type of circuit configuration is shown in Figure 11. In this circuit, a precision reference is used to drive the inverting input of an operational amplifier. The gain of the amplifier portion of the circuit is:

$$V_{OUT:AMP} = V_{IN:AMP}(1 + (R_{NTC}||R_{PAR})/R_1)$$

where:
 $V_{OUT:AMP}$ is the voltage at the output of the operational amplifier
 $V_{IN:AMP}$ is the voltage presented to the non-inverting input of the amplifier

A 2.5V precision voltage reference is used to generate the 0.276 voltage at the input to the operational amplifier. When the temperature of the NTC thermistor is equal to 0°C, the resistance of the thermistor is approximately 32,650.8Ω. The value of the parallel combination of this resistor and the 10kΩ metal film resistor (R_{PAR}) is equal to 7655.38Ω. This gives a operational amplifier gain of 14.94 V/V or an output voltage ($V_{OUT:AMP}$) of 4.093V.

When the temperature of the NTC thermistor is 50°C, the resistance of the thermistor is approximately 3601Ω. Following the same calculations above, the operational amplifier gain becomes 5.8226V/V, giving a 1.595V at the output of the amplifier.

The voltage at the output of the amplifier is used as the voltage reference of a 12-bit A/D Converter. Over the reference range of 4.093V to 1.595V the converter provides 11.75-bit accurate conversions. The converter digitizes the input signal in accordance with the transfer function:

$$DIGITAL\ OUT = \left(\frac{V_{IN:ADS}}{V_{OUT:AMP}} \right) (2^{12} - 1)$$

(to the nearest integer value)

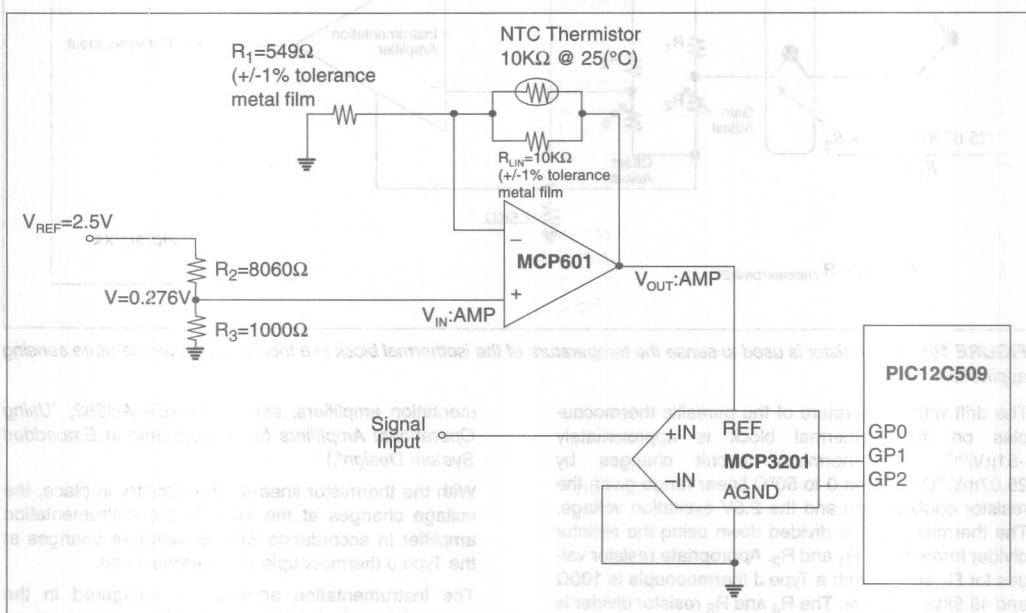
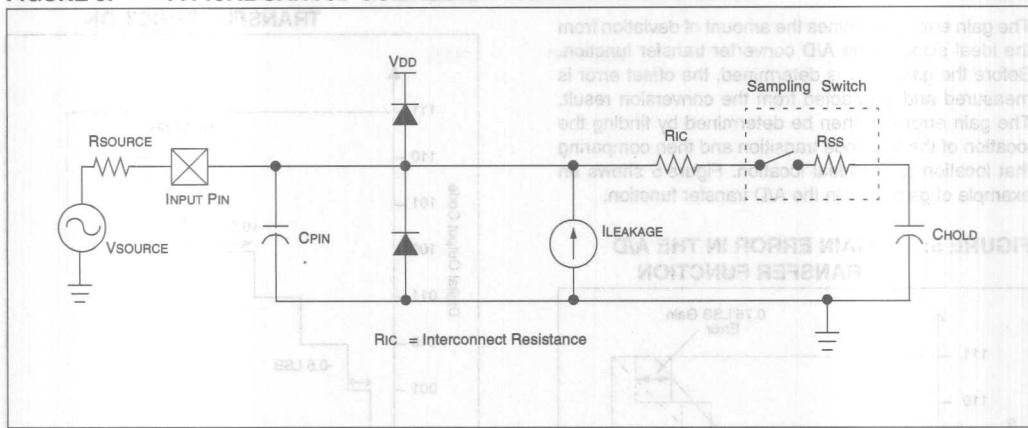


FIGURE 11: A thermistor is used to change the gain of an amplifier circuit with respect to temperature.

DS00685B-page 2-124

FIGURE 3: TYPICAL SAR A/D CONVERTER ANALOG INPUT MODEL**Conversion Time**

The conversion time is the time required to obtain the digital result after the analog input is disconnected from the holding capacitor. The conversion time is usually specified in A/D clock cycles and the minimum period for the clock is specified to obtain the specified accuracy for the A/D converter.

CODE TRANSITION POINTS

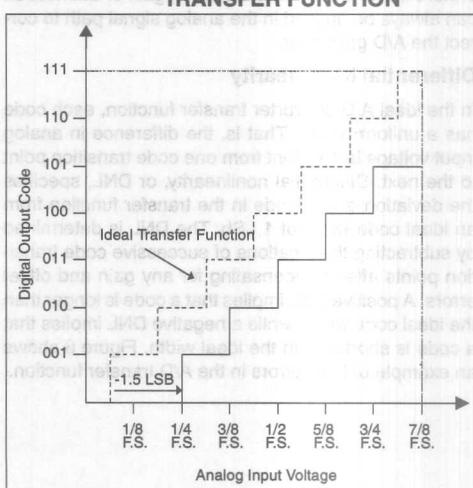
The transition points of the A/D converter are the analog input voltages at which the output code switches from one code to the next. For an ideal A/D converter, these transition points would occur at distinct, evenly spaced locations. In the real world, however, these transition points are not clearly defined due to sources of noise in the A/D converter. As an example, assume that an analog input voltage connected to the input of an A/D converter is adjusted until a constant output code is obtained. If the voltage is slowly increased or decreased from this point, there will be a range of analog input voltages that sometimes produces the first code or the next successive code in the transfer function. This range of analog inputs that produces either code is referred to as the code transition region and can be expressed statistically by averaging the results of many conversions. The code transition point is defined as the analog input level for which the probability of producing either output code is 50 percent. It is important that the code transition points are accurately determined, since the error specifications for the A/D converter are derived from them.

DC SPECIFICATIONS

The DC specifications for the A/D converter tell the designer how the device performs for steady-state analog inputs. These specifications are particularly important in instrumentation applications where the A/D converter is used to measure slowly varying physical events such as temperature, pressure or weight.

Offset Error

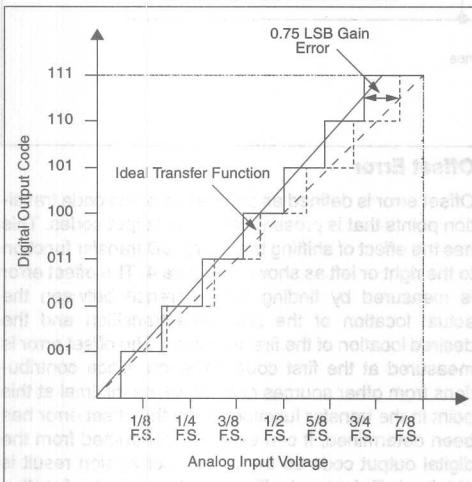
Offset error is defined as a deviation of the code transition points that is present across all output codes. This has the effect of shifting the entire A/D transfer function to the right or left as shown in Figure 4. The offset error is measured by finding the difference between the actual location of the first code transition and the desired location of the first transition. The offset error is measured at the first code transition, since contributions from other sources of error will be minimal at this point in the transfer function. Once the offset error has been determined, it can easily be subtracted from the digital output code so the correct conversion result is obtained. Referring to Figure 4, this transfer function shows that the converter has -1.5 LSB of offset error.

FIGURE 4: OFFSET ERROR IN THE A/D TRANSFER FUNCTION

Gain Error

The gain error determines the amount of deviation from the ideal slope of the A/D converter transfer function. Before the gain error is determined, the offset error is measured and subtracted from the conversion result. The gain error can then be determined by finding the location of the last code transition and then comparing that location to the ideal location. Figure 5 shows an example of gain error in the A/D transfer function.

FIGURE 5: GAIN ERROR IN THE A/D TRANSFER FUNCTION

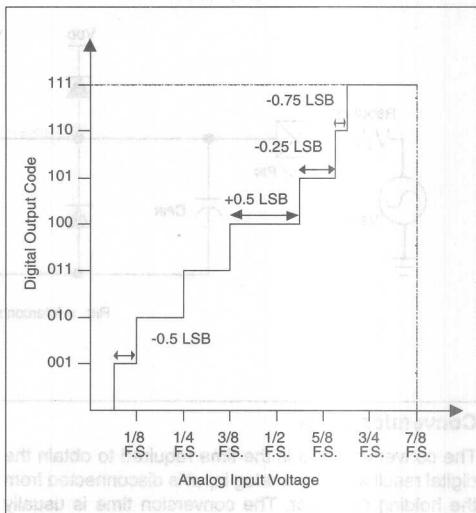


Gain error is easily compensated for in the digital measurement system by multiplying the conversion result by the necessary scaling factor. For the designer that prefers a screwdriver and trim-pots, gain or attenuation can always be applied in the analog signal path to correct the A/D gain error.

Differential Nonlinearity

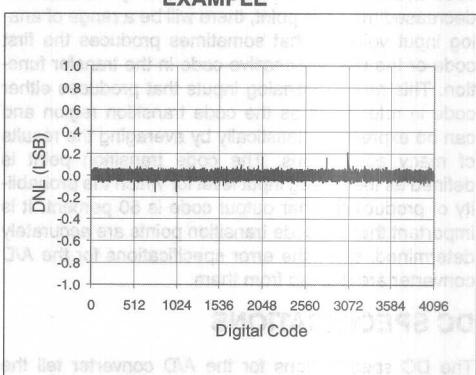
In the ideal A/D converter transfer function, each code has a uniform width. That is, the difference in analog input voltage is constant from one code transition point to the next. Differential nonlinearity, or DNL, specifies the deviation of any code in the transfer function from an ideal code width of 1 LSB. The DNL is determined by subtracting the locations of successive code transition points after compensating for any gain and offset errors. A positive DNL implies that a code is longer than the ideal code width, while a negative DNL implies that a code is shorter than the ideal width. Figure 6 shows an example of DNL errors in the A/D transfer function.

FIGURE 6: DNL ERRORS IN THE A/D TRANSFER FUNCTION



The DNL information may be provided to the designer in two ways. First, the maximum positive and negative DNL values are usually provided. Second, the DNL for each code may be provided in a graphical format. Graphical DNL data can give the designer good information regarding the 'quality' of the A/D converter. For example, a SAR A/D converter uses an array of capacitors and a comparator to determine the value of each bit in the conversion result. Imperfections of the individual capacitors will produce periodic fluctuations in the graphical DNL data. Figure 7 shows a graphical example of DNL vs. digital code.

FIGURE 7: DNL VS. DIGITAL CODE EXAMPLE



The DNL for any code cannot be less than '-1'. In fact, a DNL value of '-1' implies that a particular code does not exist at all. In other words, there is no analog input voltage that will produce the particular code.

Strictly speaking, the designer should expect that one or more codes may be missing in the transfer function if a value of -1 is specified as the minimum DNL for a particular A/D converter. However, the specifications may state that the A/D converter has a minimum DNL of -1 and will also indicate that the converter has 'no missing codes' for stated operating conditions. In this case, the minimum DNL of -1 is specified to ensure proper testing guard-bands and the probability that the designer will see a device with the actual minimum DNL near -1 is extremely low.

Integral Nonlinearity

Integral nonlinearity, or INL, is a result of cumulative DNL errors and specifies how much the overall transfer function deviates from a linear response. INL is sometimes simply referred to as the linearity of the converter. The INL specification tells the designer the best accuracy that the A/D converter will provide after calibrating the system for gain and offset. INL can be measured in two ways.

The first method used to determine INL is the end-point method. For the end-point method, the locations of the first and last code transitions for the converter are determined and a linear transfer function based on the endpoints is derived. The end-point nonlinearity is determined by finding the deviation from the derived linear transfer function at each code location.

The second method used to determine INL is the best-fit method. The best-fit response is found by manipulating the gain and offset for the measured transfer function, comparing against a linear transfer function, and balancing the total positive and negative deviations.

Figure 8 shows a comparison of linearity measurement methods for the same A/D converter. As the transfer functions indicate, the end-point method provides more conservative results, so the designer should always determine the method used to specify the INL.

The maximum positive and negative INL are usually specified for stated operating conditions. Furthermore, graphs indicating the INL for each code are sometimes given in the device data sheet. Like DNL graphical data, the INL graphical data can be used to analyze the quality of the A/D converter. Figure 9 shows a graphical example of INL vs digital code.

FIGURE 8: INL ERROR IN THE A/D TRANSFER FUNCTIONS

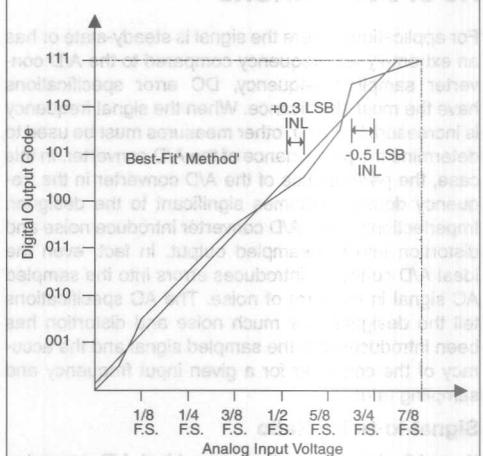
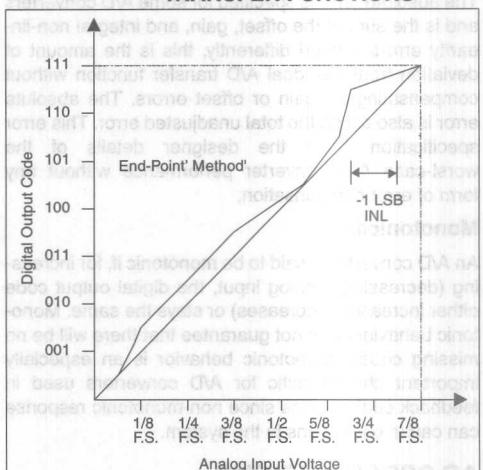
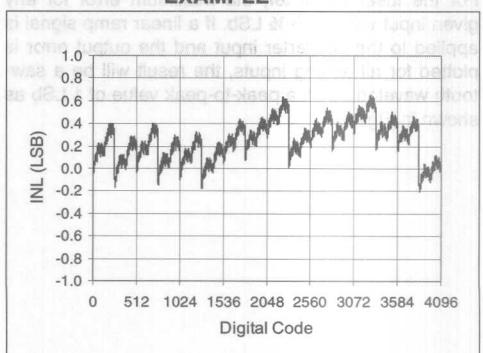


FIGURE 9: INL VS. DIGITAL CODE EXAMPLE



Absolute Error

The absolute error is specified for some A/D converters and is the sum of the offset, gain, and integral non-linearity errors. Stated differently, this is the amount of deviation from the ideal A/D transfer function without compensating for gain or offset errors. The absolute error is also called the total unadjusted error. This error specification gives the designer details of the worst-case A/D converter performance without any form of error compensation.

Monotonicity

An A/D converter is said to be monotonic if, for increasing (decreasing) analog input, the digital output code either increases (decreases) or stays the same. Monotonic behavior does not guarantee that there will be no missing codes. Monotonic behavior is an especially important characteristic for A/D converters used in feedback control loops since non-monotonic response can cause oscillations in the system.

AC SPECIFICATIONS

For applications where the signal is steady-state or has an extremely low frequency compared to the A/D converter sampling frequency, DC error specifications have the most significance. When the signal frequency is increased, however, other measures must be used to determine the performance of the A/D converter. In this case, the performance of the A/D converter in the frequency domain becomes significant to the designer. Imperfections of the A/D converter introduce noise and distortion into the sampled output. In fact, even the ideal A/D converter introduces errors into the sampled AC signal in the form of noise. The AC specifications tell the designer how much noise and distortion has been introduced into the sampled signal and the accuracy of the converter for a given input frequency and sampling rate.

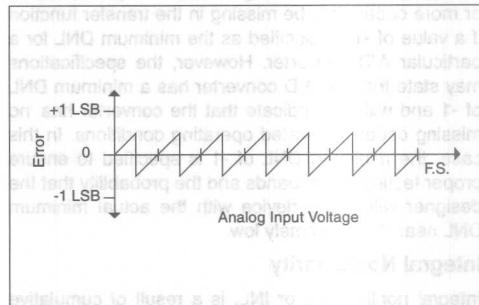
Signal-to-Noise Ratio

If an AC signal is applied to an ideal A/D converter, noise present in the digitized output will be due to quantization error.

For the ideal converter, the maximum error for any given input will be +/- 1/2 LSB. If a linear ramp signal is applied to the converter input and the output error is plotted for all analog inputs, the result will be a sawtooth waveform with a peak-to-peak value of 1 LSB as shown in Figure 10.



FIGURE 10: QUANTIZATION ERROR



The root-mean-square (RMS) amplitude of the error output can be approximated by Equation 2.

EQUATION 2: MINIMUM RMS QUANTIZATION ERROR

$$ERROR_{RMS} = \frac{1}{\sqrt{12}} \cdot 1LSB$$

The maximum theoretical signal-to-noise ratio (SNR) for an A/D converter can be determined based on the RMS quantization error determined above. If a full-scale sine wave is applied to the input of the A/D converter, the maximum theoretical SNR is given by Equation 3, where N is the resolution of the A/D converter in bits.

EQUATION 3: MAXIMUM A/D SNR

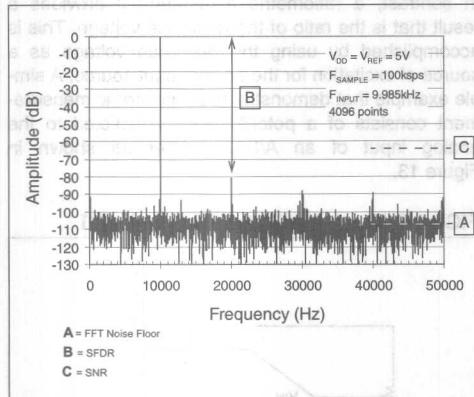
$$SNR = 6.02 \cdot N + 1.76dB$$

The above formula assumes that the signal noise is measured over the entire usable bandwidth of the A/D converter (0 - fs/2). For the case of oversampling where the signal bandwidth is less than the Nyquist bandwidth, the theoretical SNR of the A/D converter is increased by 3 dB each time the sampling frequency (fs) is doubled.

The performance of an actual A/D converter can be measured in the frequency domain by applying a sinusoidal input and performing an FFT analysis of the converter output data. Care must be taken, however, to ensure that the noise and distortion produced by the A/D converter is accurately determined. The quantization noise introduced into the sampled signal does not necessarily have a white noise spectrum and is a function of the input signal. If the sampling frequency is chosen to be an integer multiple of the signal input frequency, for example, peaks may occur in the FFT output data at harmonics of the input signal frequency due to a high degree of correlation with the quantization noise. In practice, most signals contain multiple frequencies, so the quantization noise will be randomly

dispersed throughout the FFT spectrum. Figure 11 shows example FFT data taken from a 12 bit A/D converter. Note the choices of input signal frequency and sampling frequency. You can also observe the peaks in the spectrum at harmonics of the input signal.

FIGURE 11: EXAMPLE FFT SPECTRUM FOR AN A/D CONVERTER



The FFT spectrum obtained from the A/D converter will have a noise floor that is a function of N, data resolution in bits, and M, the number of points in the FFT data. For a set of M-point FFT data, the level of the FFT noise floor can be determined using Equation 4. In Figure 11, Label 'A' indicates the level of the noise floor.

EQUATION 4: FFT NOISE FLOOR

$$6.02 \cdot N + 1.76 \text{dB} - 10 \cdot \log_{10} \left(\frac{M}{2} \right)$$

To find the actual SNR of the A/D converter, a sine wave with a level just below full-scale is applied to the input. The SNR is determined by finding the ratio of the RMS level of the input signal to the RMS value of the root-sum-square of all noise components in the FFT analysis, except for the DC component and harmonics of the input. Referring to Figure 11, Label 'C' indicates the SNR of the A/D converter. In practice, only the first several harmonics of the input are eliminated from the SNR calculation, since the higher order harmonics are usually insignificant when compared to the FFT noise floor.

Signal-to-Noise Ratio plus Distortion

The signal-to-noise ratio plus distortion, or SINAD, is measured with a sinusoidal input near full-scale applied to the A/D converter. The SINAD is found by computing the ratio of the RMS level of the input signal to the RMS value of the root-sum-square of all noise and distortion components in the FFT analysis, except for the DC component. The SINAD value is an especially useful

measure of performance, because it includes the effect of all noise, distortion and harmonics introduced by the A/D converter.

Effective Number of Bits

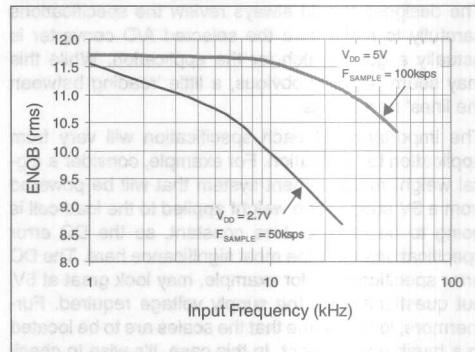
The effective number of bits (ENOB) value for an A/D converter is computed by substituting the measured SINAD value into the equation that describes the SNR for an ideal A/D converter and solving for N, the number of bits. Equation 5 shows the calculation for ENOB.

EQUATION 5: ENOB

$$\text{ENOB} = \frac{\text{SINAD} - 1.76 \text{dB}}{6.02}$$

The ENOB is usually presented for a range of input frequencies and tells the designer how accurate the converter is as a function of input frequency and the chosen sampling rate. Figure 12 shows a graphical example of ENOB data taken from an A/D converter. Note that the sampling frequency and operating conditions have been specified.

FIGURE 12: EXAMPLE ENOB DATA



Total Harmonic Distortion

The total harmonic distortion value, or THD, is the RMS value of the root-sum-square of the harmonics produced by the A/D converter relative to the RMS level of a sinusoidal input signal near full-scale. In practice, only the first several harmonics of the input signal are included in the THD measurement, because greater-order harmonics are insignificant compared to the noise floor in the measured FFT output.

Total Harmonic Distortion plus Noise

Total harmonic distortion plus noise, or THD+N, is the RMS value of the root-sum-square of the harmonics and noise produced by the A/D converter relative to the RMS level of a sinusoidal input near full-scale. THD+N does not necessarily include all data from the FFT analysis. For a valid THD+N specification, the noise bandwidth must be specified. If the noise bandwidth is taken

over the entire usable bandwidth of the A/D converter ($0 \text{ to } f_s/2$), then the THD+N measurement provides the same results as SINAD.

Spurious Free Dynamic Range

The spurious free dynamic range, or SFDR, is the ratio of the level of the input signal to the level of the largest distortion component in the FFT spectrum. This specification is important because it determines the minimum signal level that can be distinguished from distortion components. Label 'B' in Figure 11 shows the SFDR for the example A/D converter measurement data.

USING THE A/D CONVERTER

The following sections give the reader some insight on A/D measurement techniques. For more information, references to other Microchip Technology application notes have been provided at the end of this document. In addition, many other application notes that use the A/D converter are available from the Microchip Technology website.

Interpreting the Specifications

The designer should always review the specifications carefully to make sure the selected A/D converter is actually a good match for the application. While this may seem painfully obvious, a little 'reading between the lines' never hurts.

The importance of each specification will vary from application to application. For example, consider a digital weight measurement system that will be powered from a 3V supply. The weight applied to the load cell is going to be more or less constant, so the DC error specifications have the most significance here. The DC error specifications, for example, may look great at 5V, but questionable at the supply voltage required. Furthermore, let's assume that the scales are to be located in a harsh environment. In this case, it's wise to check the gain, offset and linearity specifications over the range of temperatures for which the device is expected to operate.

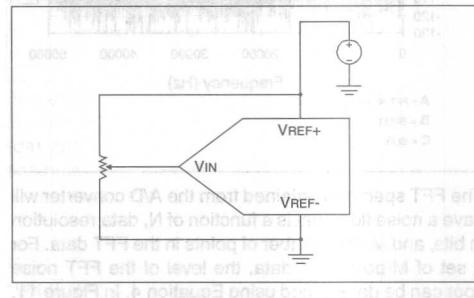
As another design example, consider an A/D converter to be used in a vibration signature analysis system. In many industrial applications, the outputs from vibration transducers attached to machinery are sampled and the data is stored in RAM for FFT analysis. By analyzing the location and amplitude of principal vibration components in the frequency domain, the machinery can be tested for faults such as cracks in the structure or worn bearings, for example. In this type of application, the AC performance parameters will have much greater significance. The SFDR, for example, will determine the minimum vibration level that can be distinguished by the A/D converter. The AC performance parameters vary over frequency, so the designer should always check the performance at the maximum frequency of interest and the desired sampling rate.

Absolute vs. Ratiometric Measurements

An absolute measurement is a measurement that compares the analog input voltage against the A/D converter reference voltage, which may be external or internal. In order for the measurement result to be accurate, the reference source must be stable over time and temperature.

In contrast, a ratiometric measurement provides a result that is the ratio of the reference voltage. This is accomplished by using the reference voltage as a source of excitation for the analog input source. A simple example that demonstrates a ratiometric measurement consists of a potentiometer connected to the analog input of an A/D converter as shown in Figure 13.

FIGURE 13: RATIO METRIC CIRCUIT



The potentiometer is connected across the power supply rails, which are also used as the reference inputs for the A/D converter. The output of the potentiometer is given by Equation 6, where x denotes the voltage division ratio of the potentiometer.

$$\text{EQUATION 6: POTENTIOMETER OUTPUT}$$

$$V_O = (V_{DD} - V_{SS}) \cdot x$$

The digital output of the A/D converter, as stated earlier, is given by Equation 7.

$$\text{EQUATION 7: DIGITAL OUTPUT OF A/D}$$

$$\text{OutputCode} = F.S. \times \frac{V_{IN}}{V_{REF}}$$

Finally, the reference voltage for the converter is given by Equation 8

EQUATION 8: REFERENCE VOLTAGE OF A/D

$$V_{REF} = V_{REF+} - V_{REF-} = V_{DD} - V_{SS}$$

If the expressions for the voltage reference and potentiometer output are substituted into the expression for the A/D output, the result is given by Equation 9.

EQUATION 9: A/D RESULT

$$\begin{aligned} OutputCode &= F.S. \times \frac{V_{IN}}{V_{REF}} \\ &= F.S. \times \frac{(V_{DD} - V_{SS}) \bullet x}{(V_{DD} - V_{SS})} \\ &= F.S. \bullet x \end{aligned}$$

This equation for the digital output shows that the ratiometric measurement is not a function of the voltage reference source. Since the conversion result only represents a percentage of full-scale, a stable reference source is not critical for accuracy of the conversion.

Performing Conversions in Sleep

All Microchip Technology microcontrollers (MCUs) that contain an A/D module have the unique ability to perform conversions with the MCU in SLEEP mode. In this mode of operation, all system operation is halted and the system oscillator is shut down to minimize the effects of digital noise on the conversion.

To perform a conversion in SLEEP, the user must select the internal A/D RC oscillator option for the A/D clock source. When the RC clock source is selected for the A/D converter, the MCU will wait one extra instruction cycle before performing the conversion so the SLEEP instruction may be executed.

One of three possible actions can occur when the conversion is finished. First, if A/D interrupts are enabled, the device will wake-up from SLEEP and continue execution at the next program instruction. Secondly, if global interrupts are also enabled on wake-up, the MCU will continue operation at the interrupt vector address. Finally, if A/D interrupts are not enabled, the A/D module will be powered down to minimize current consumption and the device will remain in SLEEP mode.

Obtaining the Best System Performance

The performance of any A/D converter can be crippled by a poor system design. It is essential, therefore, that the designer use proper analog design techniques for an application. Particular attention should be given to the power supply, grounding and PCB layout. For more information on this topic, references to other Microchip application notes are given at the end of this document.

REFERENCES FOR FURTHER READING

There are many other application notes available from the Microchip website that will provide you with technical assistance for your A/D converter application.

- AN688 - Layout Considerations for 12-bit A/D Converter Applications
- AN699 - Anti-aliasing, Analog Filters for Data Acquisition Systems
- AN719 - Interfacing Microchip's MCP3201 Analog to Digital Converter to the PICmicro® Microcontroller

NOTES: READING REFERENCES FOR THE



MICROCHIP

AN695

Interfacing Pressure Sensors to Microchip's Analog Peripherals

Author: Bonnie Baker
Microchip Technology Inc.

INTRODUCTION

Pressure measurement devices can be classified into two groups: those where pressure is the only source of power and those that require electrical excitation. The mechanical style devices that are only excited by pressure, such as bellows, diaphragms, bourdons, tubes or manometers, are usually suitable for purely mechanical systems. With these devices a change in pressure will initiate a mechanical reaction, such as a change in the position of mechanical arm or the level of liquid in a tube.

Electrically excited pressure sensors are most synergistic with the microcontroller environment. These style of sensors can be piezoresistive, Linear Variable Differential Transformers (LVDT), or capacitive sensors. Most typically, the piezoresistive sensor is used when measuring pressure.

This application note will concentrate on the signal conditioning path of the piezoresistive sensing element from sensor to microcontroller. It will show how the electrical output of this sensor can be gained, filtered and digitized in order to ready it for the microcontroller's calibration routines. This theoretical discussion will be followed with a specific pressure sensing design that is specifically designed to measure barometric pressure.

PIEZORESISTIVE PRESSURE SENSORS

2 Application Notes

The piezoresistive is a solid state, monolithic sensor that is fabricated using silicon processing. Piezo means pressure, resistance means opposition to a DC current flow. Since piezoresistive pressure sensors are fabricated on a wafer, 300 to 500 sensors can be produced per wafer. Since these wafers generate a large number of sensors they are available on the market at a reduced cost as compared to mechanical sensors.

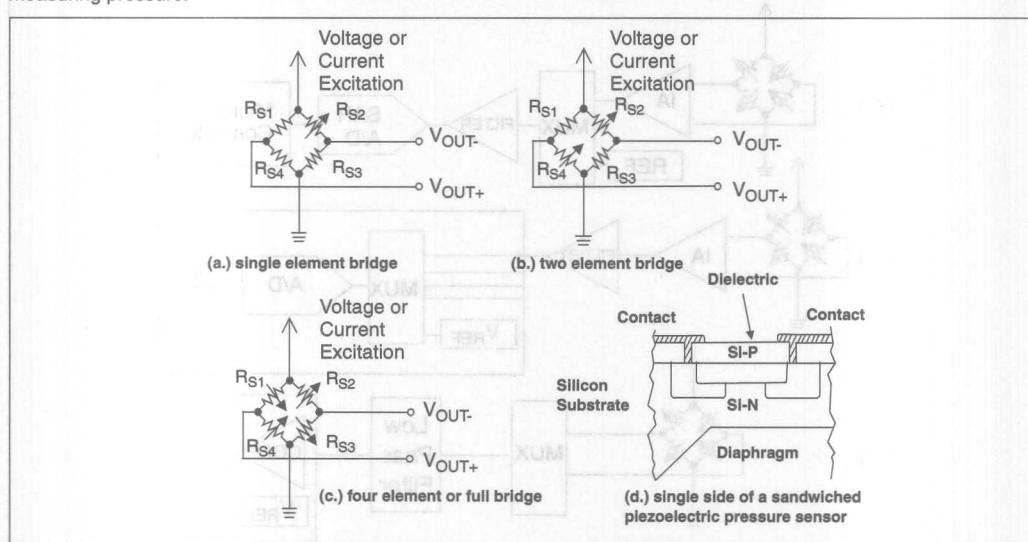


Figure 1: The resistive wheatstone bridge configuration can have one variable element (a.), two elements that vary with excitation (b.) or four elements (c.). The piezoresistive pressure sensing element is usually a four element bridge and is constructed in silicon (d.).

With this sensor, the resistors are arranged in a full wheatstone bridge configuration, which has improved sensitivity as compared to a single element or two element sensors (see Figure 1.d). When a positive differential pressure is applied to the four element bridge, two of the elements respond by compressing and the other two change to a tension state. When a negative differential pressure is applied to the sensor, the diaphragm is strained in the opposite direction and the resistors that were compressed go into a tension state, while the resistors that were in a tension state change into a compression state. Piezoresistive pressure sensors may or may not have an internal pressure reference. If they do, a pressure reference cavity is generally fabricated by sealing two wafers together. The top side of this fabricated sensor is the resistive material and the bottom is the diaphragm.

The high side of the piezoresistive bridges shown in Figure 1 can have a voltage excitation or current excitation applied. Although the magnitude of excitation (whether it is voltage or current) effects the dynamic range of the output of the sensor, the maximum difference between V_{OUT+} and V_{OUT-} generally ranges from 10s of millivolts to several hundred millivolts. The electronics that follow the sensor are used to change the differential output signal to single ended as well as gain and filter it in preparation for digitization.

ELECTRONICS SIGNAL PATH

There are several ways of capturing the small differential output signal of the sensor and transforming it into a usable digital code. One approach that can be taken is shown in the block diagram in Figure 2.a. With this approach, the small differential output of the bridge is gained and converted from differential to single ended with an instrumentation amplifier (IA). The signal may or may not travel through a multiplexer. The signal then passes through a low pass filter. The low pass filter eliminates out-of-band noise and unwanted frequencies in the system before the A/D conversion is performed. This is followed by a stand-alone A/D converter which transforms the analog signal into a usable digital code. The microcontroller takes the converter code, further calibrates and translates if need be for display purposes. In this signal path only one analog filter is required and it is positioned at the output of the multiplexer.

The second signal path shown in Figure 2.b also has an instrumentation amplifier (IA) in the signal path. Following the instrumentation amplifier stage the signal is filtered in the analog domain and then digitized with an on-chip microcontroller's A/D converter. When this type of signal path is used, every signal going into the multiplexer will require its own analog filter. Additionally, the accuracy and speed of the converter in the microcontroller is less than a stand-alone A/D converter. This may or may not be an issue in a particular application.

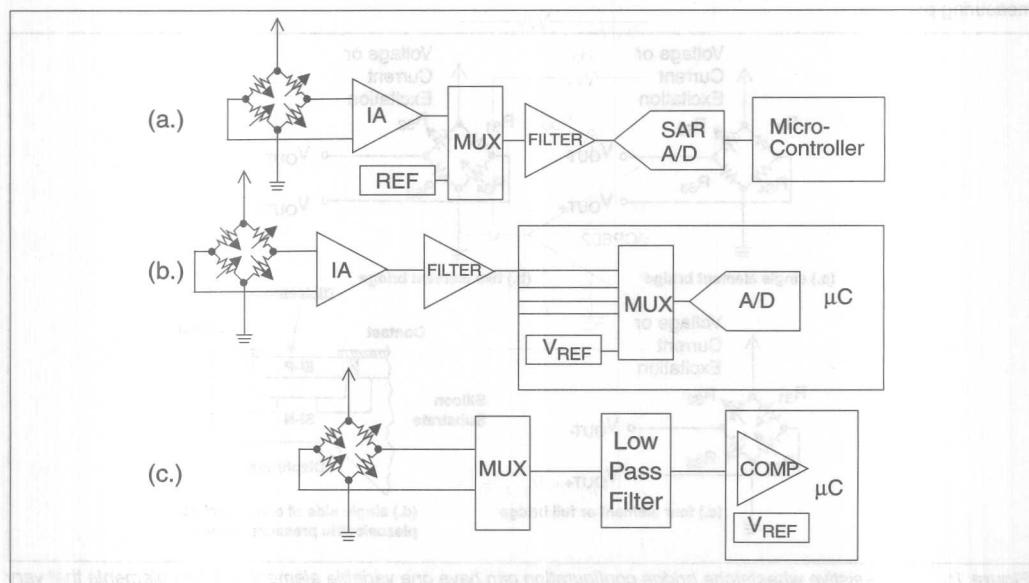


Figure 2: Three block diagrams for the piezoresistive pressure sensor signal conditioning path are shown in this Figure. The top two block diagrams, a. and b., are discussed in detail in this application note. The bottom block diagram (c.) is discussed in detail in AN717 (Microchip Technology Inc.).

INSTRUMENTATION AMPLIFIER OPTIONS AND DESIGN

With this application, the two low voltage signals from the bridge need to be subtracted in order to produce a single ended output signal. The results of this subtraction also need to be gained so that it matches the input range of the A/D converter. The implementation of the subtraction and gain functions are done so that the sensor signal is not contaminated with additional errors. The instrumentation amplifier circuits shown in Figure 3 and Figure 4 achieve all of these goals. Both of these configurations take two opposing input signals, subtract them and apply gain. The subtraction process inherently rejects common-mode voltages. Combined with these functions the signal is level shifted, making it synergistic with the signal supply environment.

The Two Op Amp Instrumentation Amplifier

A solution to the circuit problem discussed above is shown in Figure 3. The circuit in Figure 3 uses two operational amplifiers and five resistors to solve this gain and subtraction problem.

Dual amplifiers are usually used in this discrete design because of their good matching of bandwidth and over temperature performance. This instrumentation amplifier design uses the high impedance inputs of the oper-

ational amplifiers, thereby significantly reducing source impedance mismatch problems at DC. The transfer function of this circuit is equal to:

$$V_{OUT} = (V_{IN+} - V_{IN-}) \frac{R_4}{R_3} \left(1 + \frac{1}{2} \left(\frac{R_2}{R_1} + \frac{R_3}{R_4} \right) \right) + V_{CM} \left(\frac{R_4}{R_3} \left(\frac{R_3}{R_4} - \frac{R_2}{R_1} \right) \right) + V_{REF}$$

It should be noted from this transfer function that the input signals are gained along with the common-mode voltage of the two signals. The common-mode voltage can be rejected when $R_1 = R_4$ and $R_2 = R_3$. Given this change the transfer function becomes:

$$V_{OUT} = (V_{IN+} - V_{IN-}) \left(1 + \frac{R_1}{R_2} + \frac{2R_1}{R_G} \right) + V_{REF}$$

The common-mode rejection error that is caused by resistor mismatch is equal to:

$$CMR = 100 * \left(\frac{1 + \frac{R_2}{R_1}}{(\% \text{ of mismatch error})} \right)$$

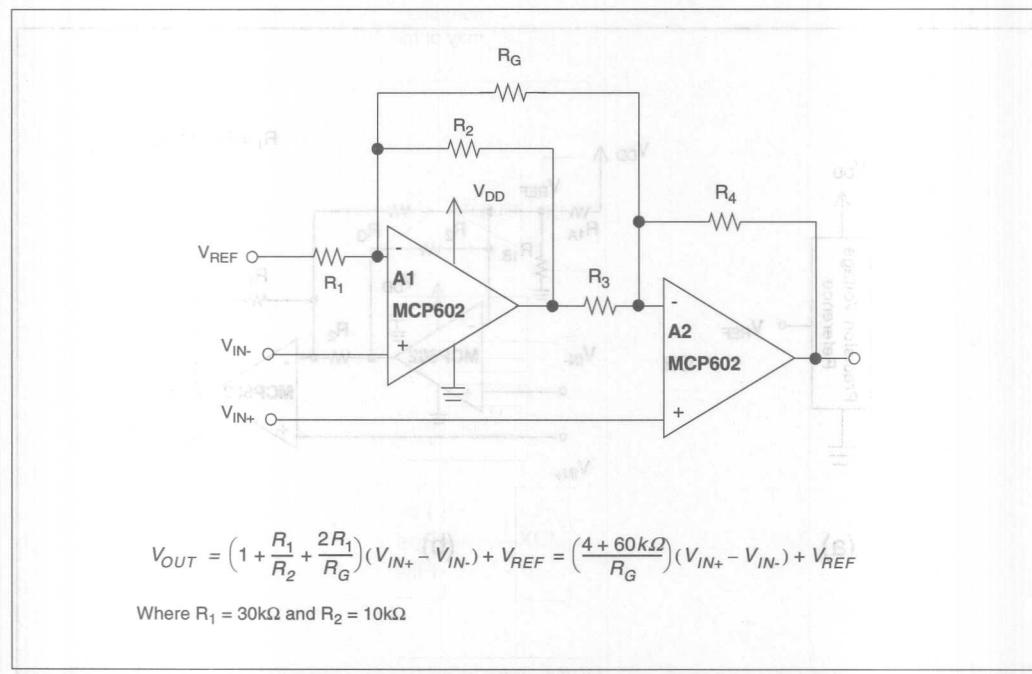


Figure 3: The two op amp instrumentation amplifier takes the difference of two input signals, gains that difference, while rejecting any voltage that is common to both of the input signals.

The ac common mode rejection for this configuration is poor. This is due to the fact that the common mode signal at V_{IN-} is inverted once with A1 and then it travels through A2 causing a second propagation delay. The common mode signal at V_{IN+} only travels through one operational amplifier (A2). Additionally, the two operational amplifiers have different closed loop gains, and consequently different closed loop bandwidths.

In terms of common-mode input range, there are two factors that limit the range of this instrumentation amplifier. The first factor involves the operation of A1 as it responds to the V_{IN-} and V_{IN+} input signals and the voltage reference, V_{REF} . The signal at the non-inverting input to A1 and A2 gained by the output of A1 by:

$$V_{OUT-A1} = V_{IN} \cdot \left(\frac{(R_G R_2 + R_1 R_2 + R_1 R_G)}{(R_1 R_2)} \right)$$

$$V_{IN+} = -V_{REF} \left(\frac{R_2}{R_1} \right)$$

The second factor that limits the common-mode input range of this circuit comes from the input swing restrictions of the amplifiers themselves.

If this circuit is in a single supply environment, it will typically require a reference that is centered at the common-mode voltage of the input signals. In Figure 4, V_{REF} serves that function. This voltage can be implemented discretely with a precision reference chip as shown in Figure 4.a or with two equal resistors in series between the power supply as shown in Figure 4.b.

Another added benefit to matching $R_2/R_1 = R_3/R_4$ is that the gain of the circuit can be changed with one resistor, R_G .

This instrumentation amplifier circuit has high impedance inputs and programmable gain capability. The features that could be improved in this circuit solution is to have the common-mode rejection independent of gain and better over frequency. These performance characteristics can only be obtained by an instrumentation amplifier configuration that has three operational amplifiers.

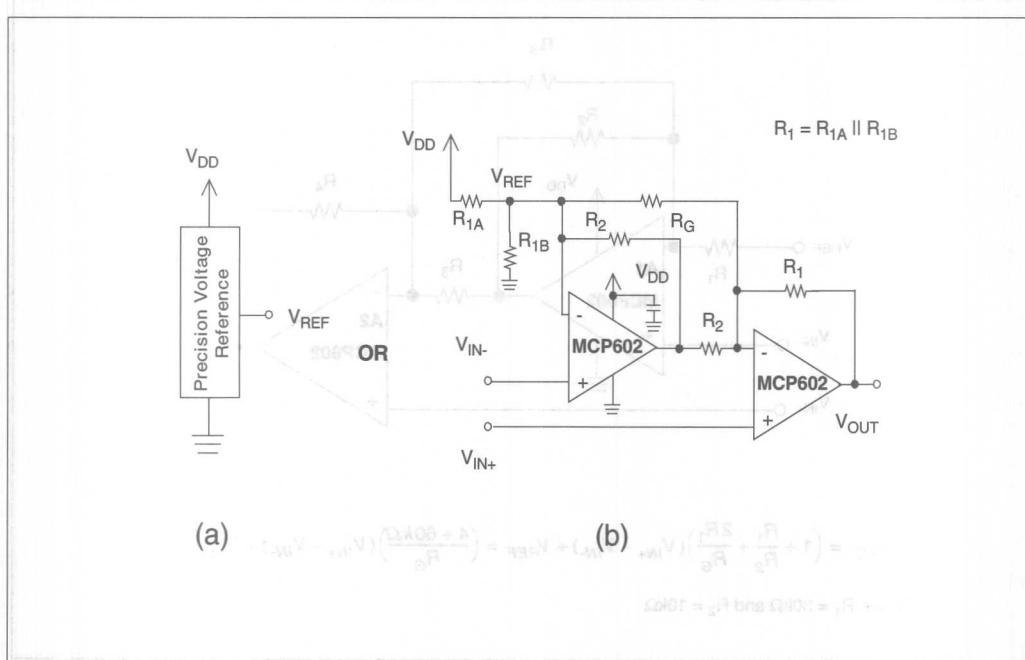


Figure 4: The reference voltage for a two op amp instrumentation amplifier in a single supply environment can be implemented with a stand-alone voltage reference (a) or a resistor divider across a voltage reference or the supply voltage (b).

The Three Op Amp Instrumentation Amplifier

An example of a more versatile instrumentation amplifier configuration is shown in Figure 5.

With this circuit configuration, two of the three amplifiers (A1 and A2) gain the two input signals. The third amplifier, A3, is used to subtract the two gained input signals, thereby providing a single ended output. The transfer function of this circuit is equal to:

$$V_{OUT} = V_{IN+} \left(1 + \frac{2R_{F2}}{R_G} \right) R_4 \left(\frac{R_1 + R_2}{(R_3 + R_4)R_1} \right) - V_{IN-} \left(1 + \frac{2R_{F1}}{R_G} \right) \left(\frac{R_2}{R_1} \right) + V_{REF} R_3 \left(\frac{R_1 + R_2}{(R_3 + R_4)R_1} \right)$$

If $R_{F2} = R_{F1}$, $R_1 = R_3$, and $R_2 = R_4$, this formula can be simplified to equal:

$$V_{OUT} = (V_{IN+} - V_{IN-}) \left(1 + \frac{2R_F}{R_G} \right) + V_{REF}$$

Quad amplifiers are typically used in the three op-amp instrumentation amplifier discrete designs because of the matching qualities of amplifiers with the same silicon. In contrast to the two op-amp instrumentation amplifier, the input signal paths (at V_{IN+} and V_{IN-}) are completely balanced. This is achieved by sending V_{IN+} and V_{IN-} signals through the same number of amplifiers to the output and using a common gain resistor, R_G .

Since this input stage is balanced, common mode currents will not flow through R_G . The common-mode rejection of this circuit is primarily dependent on the resistor matching around A3. When $R_1 = R_2 = R_3 = R_4$, common mode signals will be gained by a factor of one regardless of gain of the front end of the circuit. Consequently, large common mode signals can be handled at

all gains as long as the signals stay within A1 and A2 input and output head room limitations. If the common mode errors of the input amplifiers track they will be cancelled by the output stage.

If the assumption that R_1/R_2 equals R_3/R_4 is not correct, there could be a noticeable common mode voltage error. The calculated common-mode rejection (CMR) that is attributed to resistor mismatches in this circuit is equal to:

$$CMR = 100 * \left(\frac{1 + \frac{R_2}{R_1}}{(\% \text{ mismatch error})} \right)$$

for $R_1 = R_3$ and $R_2 = R_4$

An example of the impact of this error is demonstrated with a 12-bit, 5V system, where the gain of the circuit is 100V/V, the common-mode voltage ranges 0 to 5V and the matching error can be as large as $\pm 1\%$. Using the formula above, the contributed error of this type of common-mode excursion is equal to 1mV. This voltage is slightly less than 1LSB.

In a single supply environment, the voltage reference should be equal to the center of the input signals. This voltage is represented in the circuit in Figure 5 as V_{REF} . The purpose and effects of this reference voltage is to simply shift the output signal into the linear region of the amplifier.

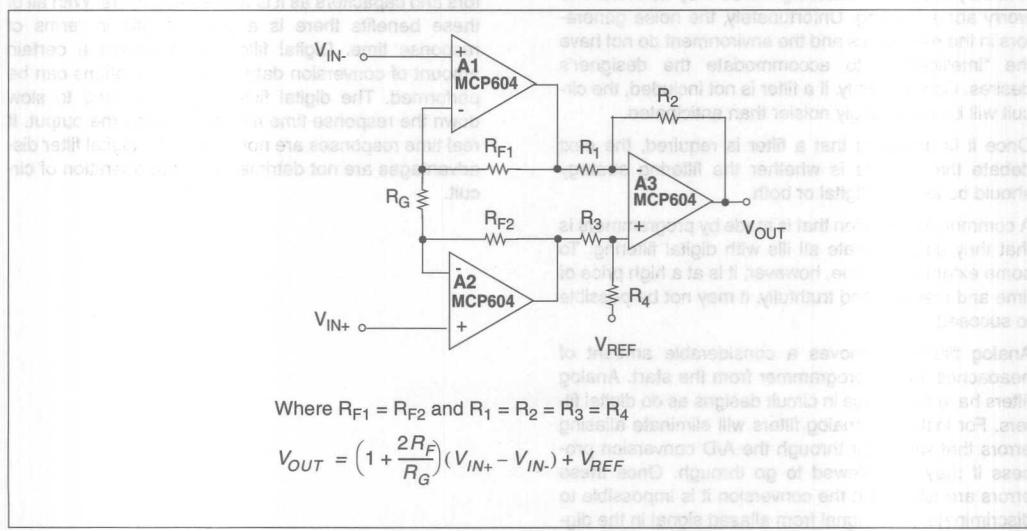


Figure 5: This is a three op amp implementation of an instrumentation amplifier. © 2000 Microchip Technology Inc.

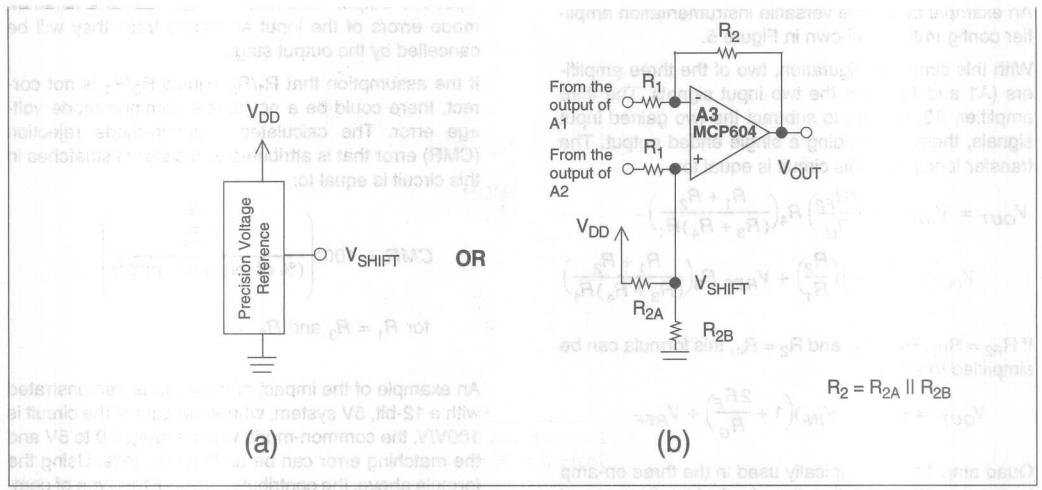


Figure 6: The reference voltage in Figure 5 can be implemented by using a precision reference circuit (a.) or a resistive voltage divider circuit (b.).

The V_{REF} circuit function can be implemented with a precision voltage reference or with the resistive network shown in Figure 6.

ANALOG FILTERING

A big topic for debate in digital design circles is whether or not an analog filter is needed and more importantly, can a digital filter replace the analog filter.

A common assumption with designers that are trying to tackle analog challenges of this type is that they claim that they are only measuring DC so they don't have to worry about filtering. Unfortunately, the noise generators in the electronics and the environment do not have the "intelligence" to accommodate the designer's desires. Consequently, if a filter is not included, the circuit will be surprisingly noisier than anticipated.

Once it is accepted that a filter is required, the next debate that ensues is whether the filtering strategy should be analog, digital or both.

A common assumption that is made by programmers is that they can eliminate all ills with digital filtering. To some extent this is true, however, it is at a high price of time and memory and truthfully, it may not be possible to succeed.

Analog filtering removes a considerable amount of headaches for the programmer from the start. Analog filters have their place in circuit designs as do digital filters. For instance, analog filters will eliminate aliasing errors that will occur through the A/D conversion process if they are allowed to go through. Once these errors are allowed in the conversion it is impossible to discriminate good signal from aliased signal in the digital domain. The analog filter also removes large signal

$$\begin{aligned}
 & \text{From the output of } A_1 \\
 & R_1 \text{ --- } V_{SHIFT} \text{ --- } + \\
 & \text{From the output of } A_2 \\
 & R_1 \text{ --- } V_{SHIFT} \text{ --- } + \\
 & \text{A3 MCP604} \text{ --- } V_{OUT} \\
 & V_{DD} \text{ --- } V_{SHIFT} \text{ --- } - \\
 & R_{2A} \text{ --- } V_{SHIFT} \text{ --- } - \\
 & R_{2B} \text{ --- } V_{OUT} \\
 & R_2 = R_{2A} \parallel R_{2B}
 \end{aligned}$$

(b)

and on hand with a basic voltage

noise that is generated by spikes or spurs in the signal. These signals are usually unintentional, but almost always destructive if not controlled. On the down side, analog filters can add to the noise floor particularly if a noisy amplifier is used with a large gain.

Where analog filters earn their worth by rejecting noise in the out of band region, digital filters can be utilized to reduce the in-band noise floor. This is implemented with oversampling algorithms. These types of filters are much easier to change on the fly because it is a matter of programming instead of a matter of changing resistors and capacitors as it is with analog filters. With all of these benefits there is a price to pay in terms of response time. Digital filters must collect a certain amount of conversion data before calculations can be performed. The digital filter algorithms tend to slow down the response time as well as delay the output. If real time responses are not critical, the digital filter disadvantages are not detrimental to the operation of circuit.



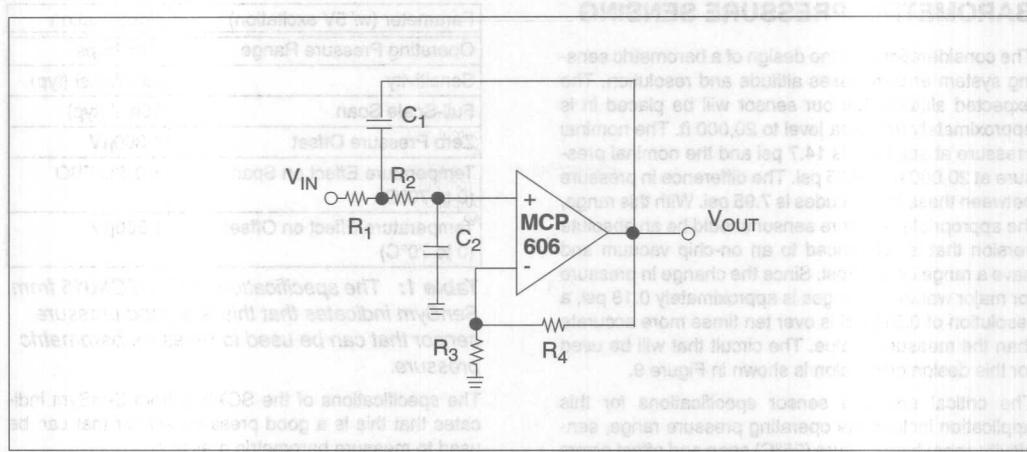


Figure 7: By using FilterLab™ software, this 2nd order low pass filter that has a non-inverting gain in the pass band can be configured as a Butterworth, Bessel, or Chebyshev filter.

As discussed previously, the hardware implementation of a low pass filter at its most fundamental level requires a capacitor and resistor for each pole. Active filters, which have one amplifier for every two poles, have the added benefit of preventing conflicting impedances and degrading the signal path.

The 2nd order lower pass filter shown in Figure 7 is one of a class of circuits that were described in 1995 by R.P Sallen and E.L. Key. With this filter the DC gain is positive. In a single supply environment this eases the implementation considerably, because a mid-supply reference is not required. This circuit not only filters high frequencies, but it can be used to gain the incoming signal.

Close inspection of this filter shows that the circuit can be configured in a gain of +1/V/V by shorting R₄ and opening R₃. In this configuration it is likely that the input of the amplifier will be exercised across a full rail-to-rail input range.

The second order Multiple Feedback circuit implementation of a low pass filter uses an amplifier, three resistors and two capacitors, as is shown in Figure 8. The DC gain of this filter is negative and easily adjusted with the ratio of R₃ and R₁. When used in a single supply environment, this circuit usually needs a voltage reference on the non-inverting input of the amplifier.

This is the filter circuit that will be used in a barometric pressure application. An adjustable voltage reference will be included in this filter design.

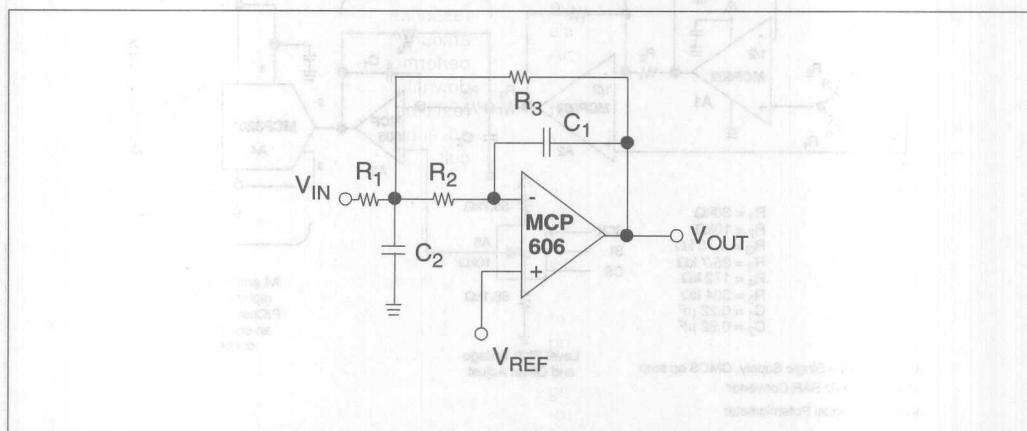


Figure 8: By using FilterLab™ software, this 2nd order low pass filter that has an inverting gain in the pass band can be configured as a Butterworth, Bessel, or Chebyshev filter.

BAROMETRIC PRESSURE SENSING

The considerations for the design of a barometric sensing system encompasses altitude and resolution. The expected altitude that our sensor will be placed in is approximately from sea level to 20,000 ft. The nominal pressure at sea level is 14.7 psi and the nominal pressure at 20,000 ft. is 6.75 psi. The difference in pressure between these two altitudes is 7.95 psi. With this range, the appropriate pressure sensor should be an absolute version that is referenced to an on-chip vacuum and have a range up to 15psi. Since the change in pressure for major weather changes is approximately 0.18 psi, a resolution of 0.015 psi is over ten times more accurate than the measured value. The circuit that will be used for this design discussion is shown in Figure 9.

The critical pressure sensor specifications for this application include the operating pressure range, sensitivity, room temperature (25°C) span and offset errors as well as over temperature (see Table 1). Although, the range of this sensor extends from 0 psi to 15 psi, this application will not be using that lower range. The minimum differential output voltage from the sensor will be 40.5mV (6.75psi or 20,000 ft.) and the maximum sensor voltage will be 88.2mV (14.7psi or sea level). The voltage at the output of the sensor is gained before it is digitized using an instrumentation amplifier.

Parameter (w/ 5V excitation)	Specification
Operating Pressure Range	0 to 15 psi
Sensitivity	6.0mV/psi (typ)
Full-Scale Span	90mV (typ)
Zero Pressure Offset	$\pm 300\mu\text{V}$
Temperature Effect on Span (0 to 70°C)	$\pm 0.5\%$ FSO
Temperature Effect on Offset (0 to 70°C)	$\pm 500\mu\text{V}$

Table 1: The specifications of the SCX015 from SenSym indicates that this is a good pressure sensor that can be used to measure barometric pressure.

The specifications of the SCX015 from SenSym indicates that this is a good pressure sensor that can be used to measure barometric pressure.

Note that temperature issues are beyond the scope of this application note. Detailed information about temperature sensing circuits can be found in Microchip's AN679, AN684, AN685, and AN687.

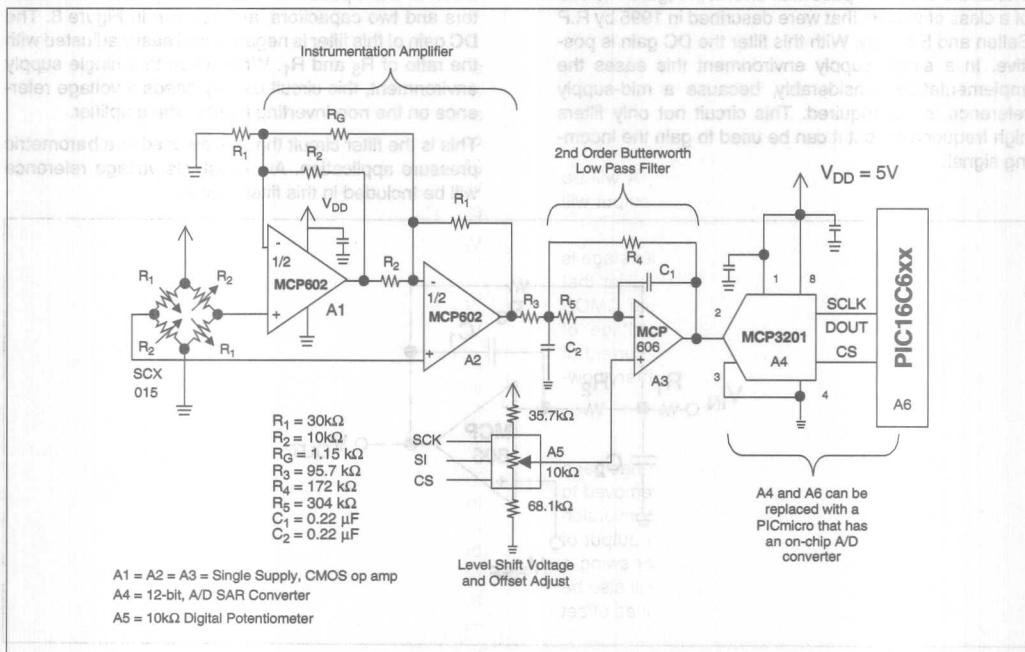


Figure 9: The voltage at the output of the SCX015 pressure sensor is gained by the instrumentation amplifier (A1 and A2) then filtered, gained and level shifted (A4) with a 2nd order low pass filter (A3) and digitized with a 12-bit A/D converter (A5).

Instrumentation Amplifier Design

This sensor requires voltage excitation. In order to determine the required gain of the circuit in Figure 9 the relationship between the maximum sensor output and allowable instrumentation amplifier output is used in the calculation. As stated previously, the maximum differential output of the sensor is 88.2mV. The allowable output range of the instrumentation amplifier is equal to $V_{DD} - 100mV$. In a five volt system where $V_{DD} = 5V$, the amplifier output maximum is equal to 4.9V. The minimum output of the sensor is 40.5mV. Since this is a positive voltage and the instrumentation amplifier is in a single supply environment, this minimum sensor output voltage will not drive the output of the instrumentation amplifier below ground. Consequently, the reference voltage called out in Figure 3 and Figure 4 is made to be equal to ground.

Gain is calculated by dividing the maximum output voltage with the maximum input voltage. Using this calculation, the appropriate gain for our system is 55.6V/V. By using the gain formula in Figure 5:

$$R_G = \frac{2R_1}{\left(\text{Gain} - 1 - \frac{R_1}{R_2}\right)}$$

If $R_1 = 30k\Omega$ and $R_2 = 10k\Omega$,

$$R_G = \frac{60k\Omega}{(55.6 - 4)} \\ = 1.15k\Omega \text{ (closest 1% value)}$$

With this gain, the maximum output of the IA will be 88.2mV*55.6V/V or 4.9V and the minimum output will be 40.5mV*55.6V/V or 2.3V.

Since the gain of this instrumentation amplifier stage is relatively large, it is desirable to use an amplifier that has a low offset voltage. The MCP607, dual CMOS amplifier has a guaranteed input offset voltage of 250 μ V (max). This amplifier's low quiescent current of 25 μ A (max) make this device attractive for battery powered applications.

Filter Design

Now that the signal from the pressure sensor has been properly differentiated and gained, noise is removed to making the results from the 12-bit A/D conversion repeatable and reliable. Remember that the output of the instrumentation amplifier circuit does not swing a full 0V to 5V. Consequently, the filter stage will also be used to implement a second gain cell as well as offset adjust.

The stop frequency of this filter is 60Hz. This will removes any mains frequencies that may be aliased back into the signal path during conversion. This being

the case, the cut-off frequency is selected to be 10Hz. Any cut-off frequency lower than 10Hz, requires capacitors that are too large, making the board implementation awkward. The total attenuation between 10Hz and 60Hz is approximately -30dB. In other words, a 60Hz signal that is part of the output signal of the instrumentation amplifier is attenuated by 0.031 times. Keeping in mind that the instrumentation amplifier has already rejected a major portion of any 60Hz common-mode signal, this level of attenuation is enough to remove any remaining 60Hz noise that exists in the signal path.

The gain and offset adjust features of this filter are also used in this segment of the application circuit. Given that the output from the instrumentation amplifier is 2.3V to 4.9V, the peak-to-peak voltage of this signal is 2.6V. A gain of 1.8V/V will produce an output swing of approximately 4.7V peak-to-peak. The adjustable offset voltage of this circuit which is gained by 2.8V/V will be configured to insure that the signal will fall at the output of the amplifier between the supplies. This adjustment circuit can also be used to remove system offset errors that originate in the sensor or instrumentation amplifier.

The filter circuit in Figure 9 can be designed with the FilterLab™ software from Microchip Technology. The two capacitors are adjusted using the FilterLab program to be equal to 0.22 μ F. This adjustment is made in order to keep the capacitor packages small enough so surface mount capacitors can be used.

The offset adjust of the filter circuit is implemented with a 10k Ω digital potentiometer in series with a 68.1k Ω and 35.7k Ω resistors. The range of the offset adjust portion of this circuit at the wiper of the digital potentiometer is from 3.0V to 3.4V. This offset circuitry is gained by the filter/amplifier circuit so that the nominal value of the offset circuitry in combination with the sensor signal is equal to:

$$\begin{aligned} V_{OUT-FILTER} &= -1.8V/V \text{ (Nominal Input Signal)} \\ &\quad + 2.8V/V \text{ (Nominal Reference voltage)} \end{aligned}$$

$$V_{OUT-FILTER} = -1.8V/V (3.6V) + 2.8V/V (3.2V)$$

$$V_{OUT-FILTER} = 2.48V$$

A key amplifier specification for this filtering circuit is input voltage noise. The MCP601, single CMOS amplifier has a typical noise density of 29 nV/ $\sqrt{\text{Hz}}$ @ 1kHz.

A/D Converter Design

The final design step for this analog signal path is to insert the analog-to digital converter. The converter quantizes a continuous analog signal into discrete buckets. The appropriate converter can be selected once it is determined how many bits the application requires.

The range of the analog signal has been closely matched to the input range of a zero to 5V in A/D converter.

The barometric pressure range is 14.7 psi to 6.75 psi. The expected increase from good weather to a strong storm system would be approximately 0.18 psi. Given this, the equipment should resolve to at least 0.015psi. This is easily achieved with a 10-bit converter. If resolution to 0.002 is needed a 12-bit a/d converter would be more suitable.

Microchip has a large variety of analog to digital converters that can be used for this application. If the stand-alone solution is appealing, the MPC320X family of 12-bit and the MCP300X 10-bit family of converters are available.

Generally speaking, stand-alone A/D converters have better accuracy than those compared to on-board converters. They also have features such a pseudo differential inputs and faster conversion speeds. The pseudo differential capability of these devices allow for configurations that reject small common mode signals. Additionally, the single channel devices can be used in simultaneous sampling applications, such as motor control. The application circuits using the single converter also require fewer analog filters because the multiplexer is typically placed before the anti-aliasing filter.

If an on-board a/d converter fits the application better, the PICmicro line has a large array of converters combined with other peripherals on a variety of micros that can be used.

The integrated solution offers a degree of flexibility that the stand-alone solution does not have. This flexibility comes in the form of operational flexibility where the device's voltage reference and sampling speed can be reconfigured on the fly. The I/O configuration is also very flexible allowing for easy implementation of the board layout.

The stand-alone and integrated A/D converters from Microchip are both suitable for the pressure sensor circuit that is shown in Figure 9.

A/D Converter Details

One of the best ways to implement a pressure sensor is to use a bridge circuit. The output of the bridge is a differential signal. This signal is converted to a single ended signal by the instrumentation amplifier. The instrumentation amplifier is configured as a non-inverting amplifier with a gain of 10. The output of the instrumentation amplifier is fed into a low pass filter. The low pass filter is a Butterworth filter with a corner frequency of 10 Hz. The output of the low pass filter is then fed into a 12-bit ADC. The ADC is configured to have a sampling rate of 100 Hz. The output of the ADC is then processed by a digital signal processor.

The digital signal processor is responsible for performing the necessary calculations to determine the pressure value.

CONCLUSION

The design challenge that has been tackled in this application note is gaining, filtering, and digitizing the small differential signal of a pressure sensor bridge. In order to achieve this goal, we used a two-op amp instrumentation amplifier which gained the differential signal from the pressure sensor and converted it to a signal ended output. After this gain stage, a 2nd order, Butterworth, anti-aliasing filter was used to reduce noise so that the A/D converter could achieve a full 10-bit accuracy.

The suggested A/D converter strategy could be on board or off board and the trade-offs were presented. Digital filtering was not needed in this application.

REFERENCES

Tandeske, Duane, Pressure Sensors, Marcel Dekker, Inc., 1991

"Anti-Aliasing Analog Filters for Data Acquisition Systems", Baker, Bonnie C., AN699, Microchip Technology Inc.

"Making a Delta-Sigma converter with a Microcontroller", Baker, Peter, Darmawaskita, Butler, AN700, Microchip Technology, Inc.

"Using Operational Amplifiers for Analog Gain in Embedded System Design", Baker, Bonnie C., AN682, Microchip Technology, Inc.

"Building a 10-bit Bridge Sensing Circuit using the PIC16C6XX and MCP601 Operational Amplifier", Baker, Bonnie C., AN717, Microchip Technology, Inc.

"Precision Temperature Sensing with RTD Circuits", Baker, Bonnie C., AN687, Microchip Technology, Inc.

"Temperature Sensing Technologies", Baker, Bonnie C., AN679, Microchip Technology, Inc.

"Thermistors in Single Supply Temperature Sensing Circuits", Baker, Bonnie C., AN685, Microchip Technology, Inc.

"Single Supply Temperature Sensing with Thermocouples", Baker, Bonnie C., AN684, Microchip Technology, Inc.

"Analog-to-Digital Converters", Microchip Application Note AN695A

"Digital-to-Analog Converters", Microchip Application Note AN695B

"Op-Amp Applications", Microchip Application Note AN695C

"Op-Amp Opener", Microchip Application Note AN695D

"Op-Amp Opener", Microchip Application Note AN695E

"Op-Amp Opener", Microchip Application Note AN695F

Anti-Aliasing, Analog Filters for Data Acquisition Systems

Author: Bonnie C. Baker
Microchip Technology Inc.

INTRODUCTION

Analog filters can be found in almost every electronic circuit. Audio systems use them for preamplification, equalization, and tone control. In communication systems, filters are used for tuning in specific frequencies and eliminating others. Digital signal processing systems use filters to prevent the aliasing of out-of-band noise and interference.

This application note investigates the design of analog filters that reduce the influence of extraneous noise in data acquisition systems. These types of systems primarily utilize low-pass filters, digital filters or a combination of both. With the analog low-pass filter, high frequency noise and interference can be removed from the signal path prior to the analog-to-digital (A/D) conversion. In this manner, the digital output code of the conversion does not contain undesirable aliased harmonic information. In contrast, a digital filter can be utilized to reduce in-band frequency noise by using averaging techniques.

Although the application note is about analog filters, the first section will compare the merits of an analog filtering strategy versus digital filtering.

Following this comparison, analog filter design parameters are defined. The frequency characteristics of a low pass filter will also be discussed with some reference to specific filter designs. In the third section, low pass filter designs will be discussed in depth.

The next portion of this application note will discuss techniques on how to determine the appropriate filter design parameters of an anti-aliasing filter. In this section, aliasing theory will be discussed. This will be followed by operational amplifier filter circuits. Examples of active and passive low pass filters will also be discussed. Finally, a 12-bit circuit design example will be given. All of the active analog filters discussed in this application note can be designed using Microchip's FilterLab software. FilterLab will calculate capacitor and resistor values, as well as, determine the number of poles that are required for the application. The program will also generate a SPICE macromodel, which can be used for spice simulations.

ANALOG VERSUS DIGITAL FILTERS

A system that includes an analog filter, a digital filter or both is shown in Figure 1. When an analog filter is implemented, it is done prior to the analog-to-digital conversion. In contrast, when a digital filter is implemented, it is done after the conversion from analog-to-digital has occurred. It is obvious why the two filters are implemented at these particular points, however, the ramifications of these restrictions are not quite so obvious.

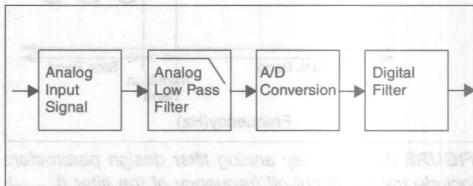


FIGURE 1: The data acquisition system signal chain can utilize analog or digital filtering techniques or a combination of the two.

There are a number of system differences when the filtering function is provided in the digital domain rather than the analog domain and the user should be aware of these.

Analog filtering can remove noise superimposed on the analog signal before it reaches the Analog-to-Digital Converter. In particular, this includes extraneous noise peaks. Digital filtering cannot eliminate these peaks riding on the analog signal. Consequently, noise peaks riding on signals near full scale have the potential to saturate the analog modulator of the A/D Converter. This is true even when the average value of the signal is within limits.

Additionally, analog filtering is more suitable for higher speed systems, i.e., above approximately 5kHz. In these types of systems, an analog filter can reduce noise in the out-of-band frequency region. This, in turn, reduces fold back signals (see the "Anti-Aliasing Filter Theory" section in this application note). The task of obtaining high resolution is placed on the A/D Converter. In contrast, a digital filter, by definition uses oversampling and averaging techniques to reduce in band and out of band noise. These two processes take time.

Since digital filtering occurs after the A/D conversion process, it can remove noise injected during the conversion process. Analog filtering cannot do this. Also,

the digital filter can be made programmable far more readily than an analog filter. Depending on the digital filter design, this gives the user the capability of programming the cutoff frequency and output data rates.

KEY LOW PASS ANALOG FILTER DESIGN PARAMETERS

A low pass analog filter can be specified with four parameters as shown in Figure 2 ($f_{\text{CUT-OFF}}$, f_{STOP} , A_{MAX} , and M).

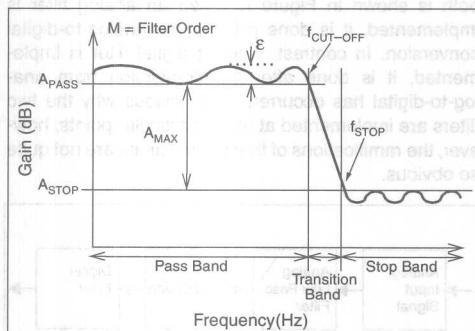


FIGURE 2: The key analog filter design parameters include the -3dB cut-off frequency of the filter ($f_{\text{cut-off}}$), the frequency at which a minimum gain is acceptable (f_{stop}) and the number of poles (M) implemented with the filter.

The cut-off frequency ($f_{\text{CUT-OFF}}$) of a low pass filter is defined as the -3dB point for a Butterworth and Bessel filter or the frequency at which the filter response leaves the error band for the Chebyshev.

The frequency span from DC to the cut-off frequency is defined as the pass band region. The magnitude of the response in the pass band is defined as A_{PASS} as shown in Figure 2. The response in the pass band can be flat with no ripple as is when a Butterworth or Bessel filter is designed. Conversely, a Chebyshev filter has a ripple up to the cut-off frequency. The magnitude of the ripple error of a filter is defined as ϵ .

By definition, a low pass filter passes lower frequencies up to the cut-off frequency and attenuates the higher frequencies that are above the cut-off frequency. An important parameter is the filter system gain, A_{MAX} . This is defined as the difference between the gain in the pass band region or $A_{\text{MAX}} = A_{\text{PASS}} - A_{\text{STOP}}$.

noise and distortion. This is done by using a digital filter to process the signal. In this case, it is a digital filter that processes the signal. The digital filter has a

In the case where a filter has ripple in the pass band, the gain of the pass band (A_{PASS}) is defined as the bottom of the ripple. The stop band frequency, f_{STOP} , is the frequency at which a minimum attenuation is reached. Although it is possible that the stop band has a ripple, the minimum gain (A_{STOP}) of this ripple is defined at the highest peak.

As the response of the filter goes beyond the cut-off frequency, it falls through the transition band to the stop band region. The bandwidth of the transition band is determined by the filter design (Butterworth, Bessel, Chebyshev, etc.) and the order (M) of the filter. The filter order is determined by the number of poles in the transfer function. For instance, if a filter has three poles in its transfer function, it can be described as a 3rd order filter.

Generally, the transition bandwidth will become smaller when more poles are used to implement the filter design. This is illustrated with a Butterworth filter in Figure 3. Ideally, a low-pass, anti-aliasing filter should perform with a "brick wall" style of response, where the transition band is designed to be as small as possible. Practically speaking, this may not be the best approach for an anti-aliasing solution. With active filter design, every two poles require an operational amplifier. For instance, if a 32nd order filter is designed, 16 operational amplifiers, 32 capacitors and up to 64 resistors would be required to implement the circuit. Additionally, each amplifier would contribute offset and noise errors into the pass band region of the response.

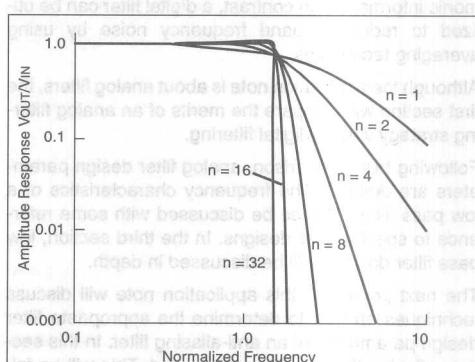


FIGURE 3: A Butterworth design is used in a low pass filter implementation to obtain various responses with frequency dependent on the number of poles or order (M) of the filter.

Strategies on how to work around these limitations will be discussed in the "Anti-Aliasing Theory" section of this application note.

ANALOG FILTER DESIGNS

The more popular filter designs are the Butterworth, Bessel, and Chebyshev. Each filter design can be identified by the four parameters illustrated in Figure 2. Other filter types not discussed in this application note include Inverse Chebyshev, Elliptic, and Cauer designs.

Butterworth Filter

The Butterworth filter is by far the most popular design used in circuits. The transfer function of a Butterworth filter consists of all poles and no zeros and is equated to:

$$V_{OUT}/V_{IN} = G/(a_0s^n + a_1s^{n-1} + a_2s^{n-2} \dots a_{n-1}s^2 + a_ns + 1)$$

where G is equal to the gain of the system.

Table 1 lists the denominator coefficients for a Butterworth design. Although the order of a Butterworth filter design theoretically can be infinite, this table only lists coefficients up to a 5th order filter.

M	a ₀	a ₁	a ₂	a ₃	a ₄
2	1.0	1.4142136			
3	1.0	2.0	2.0		
4	1.0	2.6131259	3.4142136	2.6131259	
5	1.0	3.2360680	5.2360680	5.2360680	3.2360680

TABLE 1: Coefficients versus filter order for Butterworth designs.

As shown in Figure 4a., the frequency behavior has a maximally flat magnitude response in pass-band. The rate of attenuation in transition band is better than Bessel, but not as good as the Chebyshev filter. There is no ringing in stop band. The step response of the Butterworth is illustrated in Figure 5a. This filter type has some overshoot and ringing in the time domain, but less than the Chebyshev.

Chebyshev Filter

The transfer function of the Chebyshev filter is only similar to the Butterworth filter in that it has all poles and no zeros with a transfer function of:

$$V_{OUT}/V_{IN} = G/(a_0 + a_1s + a_2s^2 + \dots a_{n-1}s^{n-1} + s_n)$$

Its frequency behavior has a ripple (Figure 4b.) in the pass-band that is determined by the specific placement of the poles in the circuit design. The magnitude of the ripple is defined in Figure 2 as ϵ . In general, an increase in ripple magnitude will lessen the width of the transition band.

The denominator coefficients of a 0.5dB ripple Chebyshev design are given in Table 2. Although the order of a Chebyshev filter design theoretically can be infinite, this table only lists coefficients up to a 5th order filter.

M	a ₀	a ₁	a ₂	a ₃	a ₄
2	1.516203	1.425625			
3	0.715694	1.534895	1.252913		
4	0.379051	1.025455	1.716866	1.197386	
5	0.178923	0.752518	1.309575	1.937367	1.172491

TABLE 2: Coefficients versus filter order for 1/2dB ripple Chebyshev designs.

The rate of attenuation in the transition band is steeper than Butterworth and Bessel filters. For instance, a 5th order Butterworth response is required if it is to meet the transition band width of a 3rd order Chebyshev. Although there is ringing in the pass band region with this filter, the stop band is void of ringing. The step response (Figure 5b.) has a fair degree of overshoot and ringing.

Bessel Filter

Once again, the transfer function of the Bessel filter has only poles and no zeros. Where the Butterworth design is optimized for a maximally flat pass band response and the Chebyshev can be easily adjusted to minimize the transition bandwidth, the Bessel filter produces a constant time delay with respect to frequency over a large range of frequency. Mathematically, this relationship can be expressed as:

$$C = -\Delta\theta * \Delta f$$

where:

C is a constant,

θ is the phase in degrees, and

f is frequency in Hz

Alternatively, the relationship can be expressed in degrees per radian as:

$$C = -\Delta\theta / \Delta\omega$$

where:

C is a constant,

θ is the phase in degrees, and

ω is in radians.

The transfer function for the Bessel filter is:

$$V_{OUT}/V_{IN} = G/(a_0 + a_1s + a_2s^2 + \dots a_{n-1}s^{n-1} + s_n)$$

The denominator coefficients for a Bessel filter are given in Table 3. Although the order of a Bessel filter design theoretically can be infinite, this table only lists coefficients up to a 5th order filter.

M	a ₀	a ₁	a ₂	a ₃	a ₄
2	3	3	3		
3	15	15	6		
4	105	105	45	10	
5	945	945	420	105	15

TABLE 3: Coefficients versus filter order for Bessel designs.

The Bessel filter has a flat magnitude response in pass-band (Figure 4c). Following the pass band, the rate of attenuation in transition band is slower than the Butterworth or Chebyshev. And finally, there is no ringing in stop band. This filter has the best step response of all the filters mentioned above, with very little overshoot or ringing (Figure 5c.).

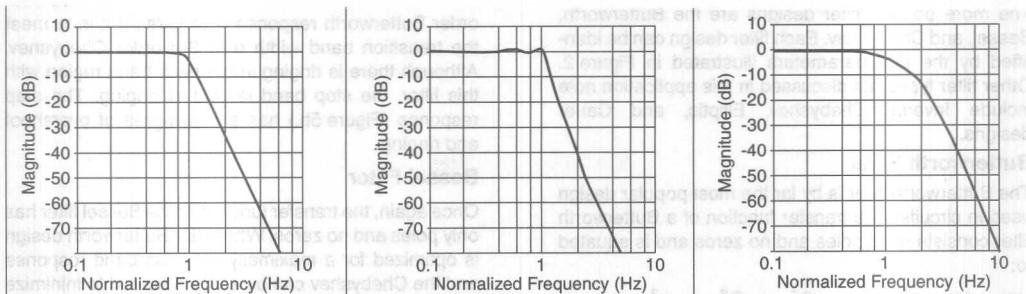


FIGURE 4: The frequency responses of the more popular filters, Butterworth (a), Chebyshev (b), and Bessel (c).

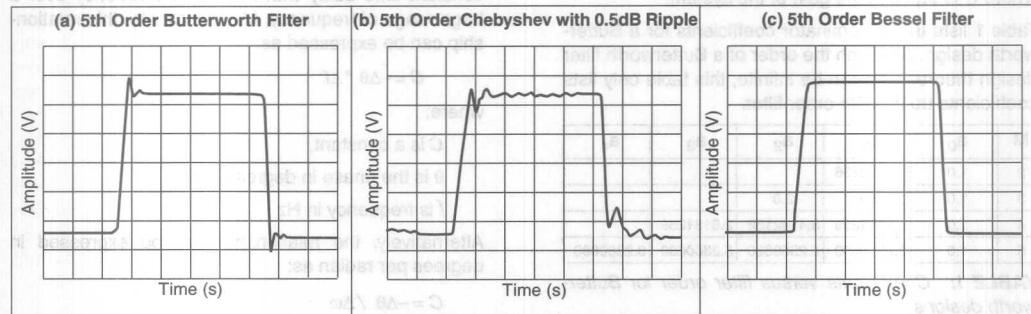


FIGURE 5: The step response of the 5th order filters shown in Figure 4 are illustrated here.

ANTI-ALIASING FILTER THEORY

A/D Converters are usually operated with a constant sampling frequency when digitizing analog signals. By using a sampling frequency (f_s), typically called the Nyquist rate, all input signals with frequencies below $f_s/2$ are reliably digitized. If there is a portion of the input signal that resides in the frequency domain above $f_s/2$, that portion will fold back into the bandwidth of interest with the amplitude preserved. The phenomena makes it impossible to discern the difference between a signal from the lower frequencies (below $f_s/2$) and higher frequencies (above $f_s/2$).

This aliasing or fold back phenomena is illustrated in the frequency domain in Figure 6.

In both parts of this figure, the x-axis identifies the frequency of the sampling system, f_s . In the left portion of Figure 6, five segments of the frequency band are identified. Segment $N = 0$ spans from DC to one half of the sampling rate. In this bandwidth, the sampling system will reliably record the frequency content of an analog input signal. In the segments where $N > 0$, the frequency content of the analog signal will be recorded by the digitizing system in the bandwidth of the segment $N = 0$. Mathematically, these higher frequencies will be folded back with the following equation:

$$f_{\text{ALIASED}} = |f_{IN} - Nf_s|$$

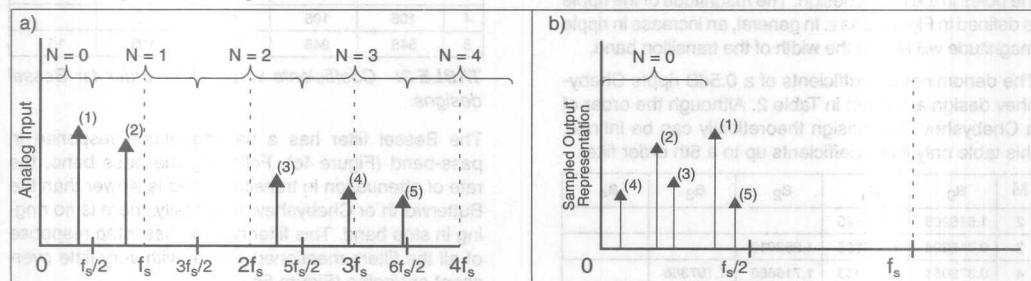


FIGURE 6: A system that is sampling an input signal at f_s (a) will identify signals with frequencies below $f_s/2$ as well as above. Input signals below $f_s/2$ will be reliably digitized while signals above $f_s/2$ will be folded back (b) and appear as lower frequencies in the digital output.

For example, let the sampling rate, (f_s), of the system be equal to 100kHz and the frequency content of:

$$f_{IN}(1) = 41\text{kHz}$$

$$f_{IN}(2) = 82\text{kHz}$$

$$f_{IN}(3) = 219\text{kHz}$$

$$f_{IN}(4) = 294\text{kHz}$$

$$f_{IN}(5) = 347\text{kHz}$$

The sampled output will contain accurate amplitude information of all of these input signals, however, four of them will be folded back into the frequency range of DC to $f_s/2$ or DC to 50kHz. By using the equation $f_{OUT} = f_{IN} - Nf_s$, the frequencies of the input signals are transformed to:

$$f_{OUT}(1) = 41\text{kHz} - 0 \times 100\text{kHz} = 41\text{kHz}$$

$$f_{OUT}(2) = 82\text{kHz} - 1 \times 100\text{kHz} = 18\text{kHz}$$

$$f_{OUT}(3) = 219\text{kHz} - 2 \times 100\text{kHz} = 19\text{kHz}$$

$$f_{OUT}(4) = 294\text{kHz} - 3 \times 100\text{kHz} = 6\text{kHz}$$

$$f_{OUT}(5) = 347\text{kHz} - 4 \times 100\text{kHz} = 53\text{kHz}$$

Note that all of these signal frequencies are between DC and $f_s/2$ and that the amplitude information has been reliably retained.

This frequency folding phenomena can be eliminated or significantly reduced by using an analog low pass filter prior to the A/D Converter input. This concept is illustrated in Figure 7. In this diagram, the low pass filter attenuates the second portion of the input signal at frequency (2). Consequently, this signal will not be aliased into the final sampled output. There are two regions of the analog low pass filter illustrated in Figure 7. The region to the left is within the bandwidth of DC to $f_s/2$. The second region, which is shaded, illustrates the transition band of the filter. Since this region is greater than $f_s/2$, signals within this frequency band will be aliased into the output of the sampling system. The effects of this error can be minimized by moving the corner frequency of the filter lower than $f_s/2$ or increasing the order of the filter. In both cases, the minimum gain of the filter, A_{STOP} , at $f_s/2$ should less than the signal-to-noise ratio (SNR) of the sampling system.

For instance, if a 12-bit A/D Converter is used, the ideal SNR is 74dB. The filter should be designed so that its gain at f_{STOP} is at least 74dB less than the pass band gain. Assuming a 5th order filter is used in this example:

$$f_{CUT-OFF} = 0.18f_s/2 \text{ for a Butterworth Filter}$$

$$f_{CUT-OFF} = 0.11f_s/2 \text{ for a Bessel Filter}$$

$$f_{CUT-OFF} = 0.21f_s/2 \text{ for a Chebyshev Filter with } 0.5\text{dB ripple in the pass band}$$

$$f_{CUT-OFF} = 0.26f_s/2 \text{ for a Chebyshev Filter with } 1\text{dB ripple in the pass band}$$

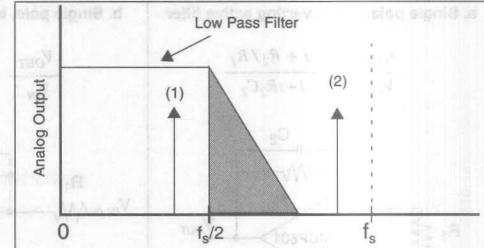


FIGURE 7: If the sampling system has a low pass analog filter prior to the sampling mechanism, high frequency signals will be attenuated and not sampled.

ANALOG FILTER REALIZATION

Traditionally, low pass filters were implemented with passive devices, ie. resistors and capacitors. Inductors were added when high pass or band pass filters were needed. At the time active filter designs were realizable, however, the cost of operational amplifiers was prohibitive. Passive filters are still used with filter design when a single pole filter is required or where the bandwidth of the filter operates at higher frequencies than leading edge operational amplifiers. Even with these two exceptions, filter realization is predominately implemented with operational amplifiers, capacitors and resistors.

Passive Filters

Passive, low pass filters are realized with resistors and capacitors. The realization of single and double pole low pass filters are shown in Figure 8.

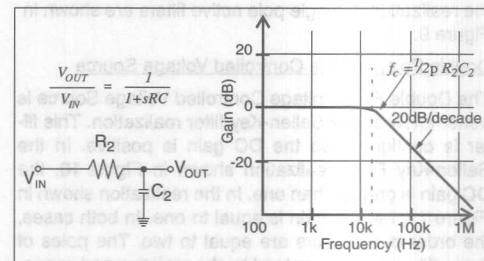


FIGURE 8: A resistor and capacitor can be used to implement a passive, low pass analog filter. The input and output impedance of this type of filter implementation is equal to R_2 .

The output impedance of a passive low pass filter is relatively high when compared to the active filter realization. For instance, a 1kHz low pass filter which uses a $0.1\mu\text{F}$ capacitor in the design would require a $1.59\text{k}\Omega$ resistor to complete the implementation. This value of resistor could create an undesirable voltage drop or make impedance matching difficult. Consequently, passive filters are typically used to implement a single pole. Single pole operational amplifier filters have the added benefit of "isolating" the high impedance of the filter from the following circuitry.

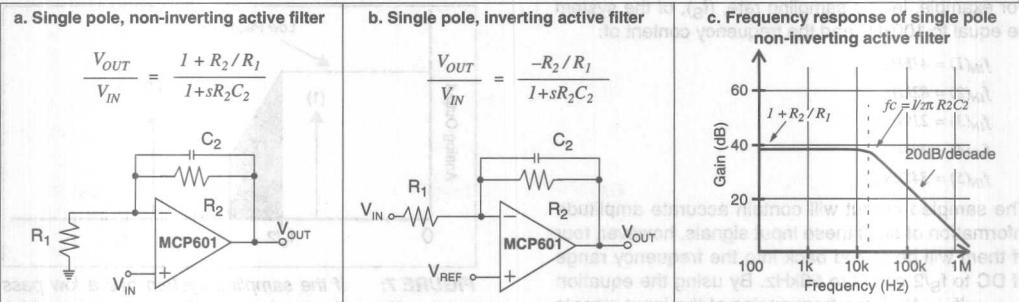


FIGURE 9: An operational amplifier in combination with two resistors and one capacitor can be used to implement a 1st order filter. The frequency response of these active filters is equivalent to a single pole passive low pass filter.

It is very common to use a single pole, low pass, passive filter at the input of a Delta-Sigma A/D Converter. In this case, the high output impedance of the filter does not interfere with the conversion process.

Active Filters

An active filter uses a combination of one amplifier, one to three resistors and one to two capacitors to implement one or two poles. The active filter offers the advantage of providing "isolation" between stages. This is possible by taking advantage of the high input impedance and low output impedance of the operational amplifier. In all cases, the order of the filter is determined by the number of capacitors at the input and in the feedback loop of the amplifier.

Single Pole Filter

The frequency response of the single pole, active filter is identical to a single pole passive filter. Examples of the realization of single pole active filters are shown in Figure 9.

Double Pole Voltage Controlled Voltage Source

The Double Pole, Voltage Controlled Voltage Source is better known as the Sallen-Key filter realization. This filter is configured so the DC gain is positive. In the Sallen-Key Filter realization shown in Figure 10, the DC gain is greater than one. In the realization shown in Figure 11, the DC gain is equal to one. In both cases, the order of the filters are equal to two. The poles of these filters are determined by the resistive and capacitive values of R_1 , R_2 , C_1 , and C_2 .

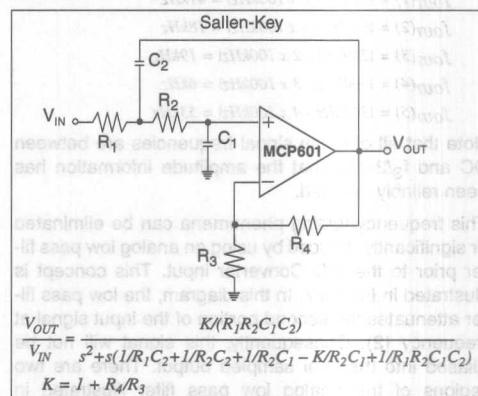


FIGURE 10: The double pole or Sallen-Key filter implementation has a gain $G = 1 + R_4/R_3$. If R_3 is open and R_4 is shorted the DC gain is equal to $1/V_V$.

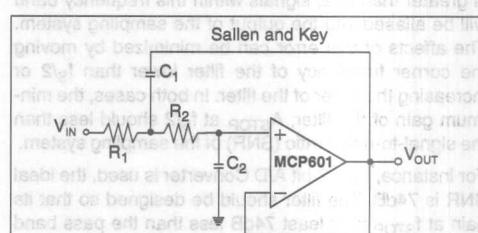


FIGURE 11: The double pole or Sallen-Key filter implementation with a DC gain is equal to $1/V/V$.

Double Pole Multiple Feedback

The double pole, multiple feedback realization of a 2nd order low pass filter is shown in Figure 12. This filter can also be identified as simply a Multiple Feedback Filter. The DC gain of this filter inverts the signal and is equal to the ratio of R_1 and R_2 . The poles are determined by the values of R_1 , R_3 , C_1 , and C_2 .

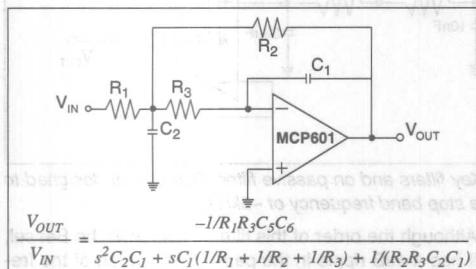


FIGURE 12: A double pole, multiple feedback circuit implementation uses three resistors and two capacitors to implement a 2nd order analog filter. DC gain is equal to $-R_2 / R_1$.

ANTI-ALIASING FILTER DESIGN EXAMPLE

In the following examples, the data acquisition system signal chain shown in Figure 1 will be modified as follows. The analog signal will go directly into an active low pass filter. In this example, the bandwidth of interest of the analog signal is DC to 1kHz. The low pass filter will be designed so that high frequency signals from the analog input do not pass through to the A/D Converter in an attempt to eliminate aliasing errors. The implementation and order of this filter will be modified according to the design parameters. Excluding the filtering function, the anti-aliasing filter will not modify the signal further, i.e., implement a gain or invert the signal. The low pass filter segment will be followed by a 12-bit SAR A/D Converter. The sampling rate of the A/D Converter will be 20kHz, making 1/2 of Nyquist equal to 10kHz. The ideal signal-to-noise ratio of a 12-bit A/D Converter of 74dB. This design parameter will be used when determining the order of the anti-aliasing filter. The filter examples discussed in this section were generated using Microchip's FilterLab software.

Three design parameters will be used to implement appropriate anti-aliasing filters:

1. Cut-off frequency for filter must be 1kHz or higher.
2. Filter attenuates the signal to -74dB at 10kHz.
3. The analog signal will only be filtered and not gained or inverted.

Implementation with Bessel Filter Design

A Bessel Filter design is used in Figure 13 to implement the anti-aliasing filter in the system described above. A 5th order filter that has a cut-off frequency of 1kHz is required for this implementation. A combination of two Sallen-Key filters plus a passive low pass filter are designed into the circuit as shown in Figure 14. This filter attenuates the analog input signal 79dB from the pass band region to 10kHz. The frequency response of this Bessel, 5th order filter is shown in Figure 13.

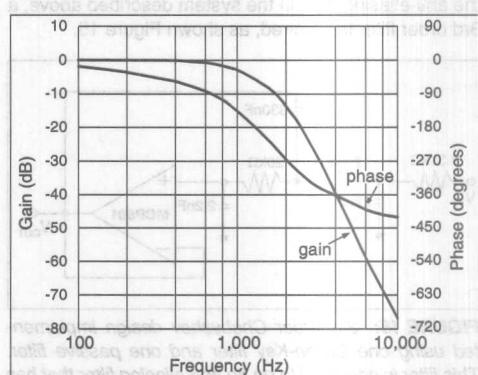


FIGURE 13: Frequency response of 5th order Bessel design implemented in Figure 14.

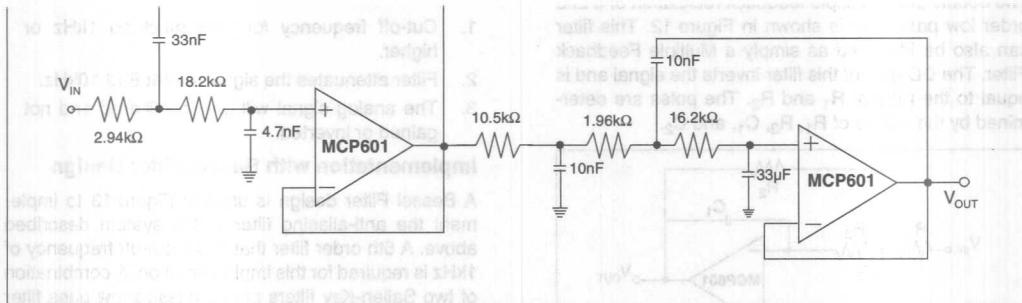


FIGURE 14: 5th order Bessel design implemented two Sallen-Key filters and one passive filter. This filter is designed to be an anti-aliasing filter that has a cut-off frequency of 1kHz and a stop band frequency of ~5kHz.

Implementation with Chebyshev Design

When a Chebyshev filter design is used to implement the anti-aliasing filter in the system described above, a 3rd order filter is required, as shown Figure 15.

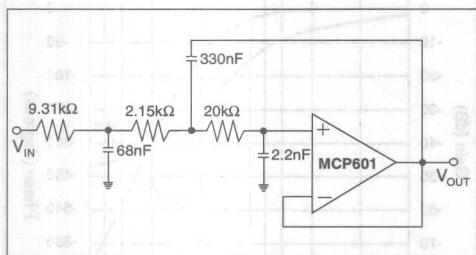
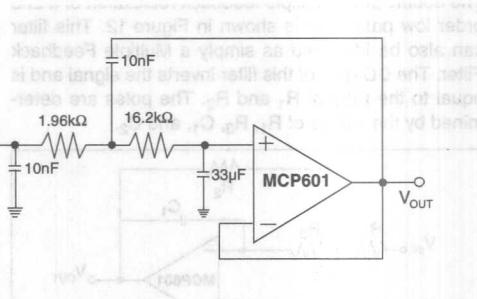


FIGURE 15: 3rd order Chebyshev design implemented using one Sallen-Key filter and one passive filter. This filter is designed to be an anti-aliasing filter that has a cut-off frequency of 1kHz -4db ripple and a stop band frequency of ~5kHz.



Although the order of this filter is less than the Bessel, it has a 4dB ripple in the pass band portion of the frequency response. The combination of one Sallen-Key filter plus a passive low pass filter is used. This filter is attenuated to -70dB at 10kHz. The frequency response of this Chebyshev 3rd order filter is shown in Figure 16.

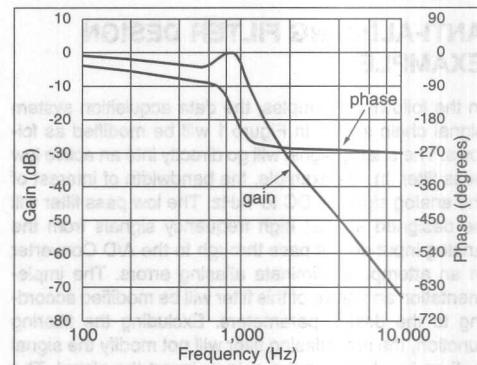


FIGURE 16: Frequency response of 3rd order Chebyshev design implemented in Figure 15.

This filter provides less than the ideal 74dB of dynamic range (A_{MAX}), which should be taken into consideration.

The difference between -70dB and -74dB attenuation in a 12-bit system will introduce little less than 1/2 LSB error. This occurs as a result of aliased signals from 10kHz to 11.8kHz. Additionally, a 4dB gain error will occur in the pass band. This is a consequence of the ripple response in the pass band, as shown in Figure 16.

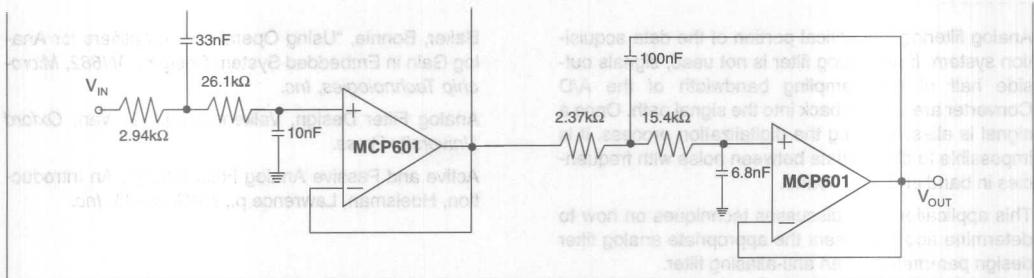


FIGURE 17: 4th order Butterworth design implemented two Sallen-Key filters. This filter is designed to be an anti-aliasing filter that has a cut-off frequency of 1kHz and a stop band frequency of ~5kHz.

Implementation with Butterworth Design

As a final alternative, a Butterworth filter design can be used in the filter implementation of the anti-aliasing filter, as shown in Figure 17.

For this circuit implementation, a 4th order filter is used with a cut-off frequency of 1kHz. Two Sallen-Key filters are used. This filter attenuates the pass band signal 80dB at 10kHz. The frequency response of this Butterworth 4th order filter is shown in Figure 18.

The frequency response of the three filters described above along with several other options are summarized in Table 4.

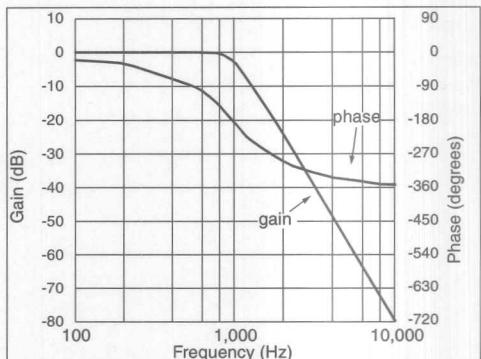


FIGURE 18: Frequency response of 4th order Butterworth design implemented in Figure 17.

FILTER ORDER, M	BUTTERWORTH, A_{MAX} (dB)	BESSEL, A_{MAX} (dB)	CHEBYSHEV, A_{MAX} (dB) W/ RIPPLE ERROR OF 1dB	CHEBYSHEV, A_{MAX} (dB) W/ RIPPLE ERROR OF 4dB
3	60	51	65	70
4	80	66	90	92
5	100	79	117	122
6	120	92	142	144
7	140	104	169	174

TABLE 4: Theoretical frequency response at 10kHz of various filter designs versus filter order. Each filter has a cut-off frequency of 1kHz.

CONCLUSION

Analog filtering is a critical portion of the data acquisition system. If an analog filter is not used, signals outside half of the sampling bandwidth of the A/D Converter are aliased back into the signal path. Once a signal is aliased during the digitalization process, it is impossible to differentiate between noise with frequencies in band and out of band.

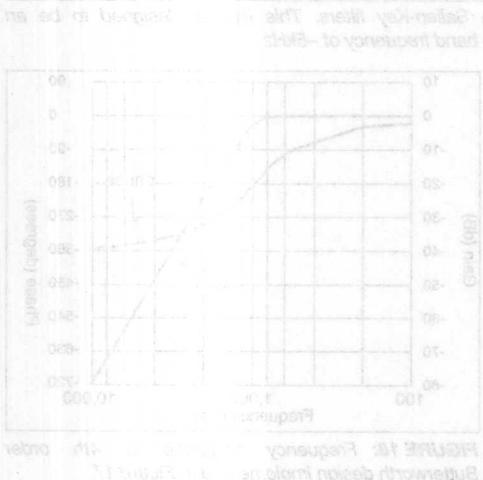
This application note discusses techniques on how to determine and implement the appropriate analog filter design parameters of an anti-aliasing filter.

REFERENCES

Baker, Bonnie, "Using Operational Amplifiers for Analog Gain in Embedded System Design", AN682, *Microchip Technologies, Inc.*

Analog Filter Design, Valkenburg, M. E. Van, *Oxford University Press*.

Active and Passive Analog Filter Design, An Introduction, Huelsman, Lawrence p., *McGraw Hill, Inc.*



(8b) XAM A FILTER ORDER	(8b) XAM A FILTER ORDER	XAM FILTER ORDER (8b)	HTROWRET (8b) XAM	RETUR NEGRO M
05	00	05	00	5
06	00	06	00	4
07	01	07	00	3
08	01	08	00	2
09	00	09	00	1

DS00699B-page 2-170
© 1999 Microchip Technology Inc.



MICROCHIP

AN702

Interfacing Microchip MCP3201 A/D Converter to 8051-Based Microcontroller

Author: Lee Studley
Microchip Technology Inc.

INTRODUCTION

In embedded controller applications, it is often desirable to provide a means to digitize analog signals. The MCP3201 12-bit Analog-to-Digital (A/D) Converter gives the designer an easy means to add this feature to a microcontroller with a minimal number of connections.

This Application Note will demonstrate how easy it is to connect the MPC3201 to an 8051-compatible microprocessor.

The MCP3201 is a fast 100kHz 12-bit A/D Converter featuring low power consumption and power saving standby modes. The features of the device include an onboard sample-hold and a single pseudo differential input. Output data from the MCP3201 is provided by a high speed serial interface that is compatible with the SPI™ protocol. The MCP3201 operates over a broad voltage range (2.7V – 5.5V). The device is offered in 8-pin PDIP and 150mil SOIC packages.

The MCP3201 connects to the target microprocessor via an SPI-like serial interface that can be controlled by I/O commands, or by using the synchronous resources commonly found in microcontrollers. Two methods will be explored in supporting the serial format for the A/D Converter: An I/O port "bit-banging" method and a method that uses the 8051 UART in synchronous serial mode 0. An 8051 derivative processor, the 80C320, was chosen for testing since it has a second onboard serial port. This second serial port allows the A/D Converter sample data to be echoed to a host PC running an ASCII terminal program such as Hyperterm. Both ports respond to the standard 8051 setup instructions for code portability. An 8051 has a single UART that can be dedicated to either the A/D Converter, or to other communication tasks.

I/O PORT METHOD

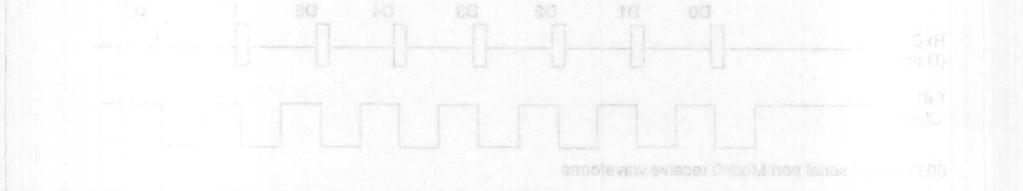
The serial data format supported by the MCP3201 is illustrated in Figure 1. The A/D Converter will come out of its sleep mode on the falling edge of CS. The conversion is then initiated with the first rising edge of CLK. During the next 1.5 CLK cycles, the converter samples the input signal. The sampling period stops at the end of the 1.5 CLK cycles on the falling edge of CLK, and D_{OUT} also changes from a Hi-Z state to null. Following the transmission of the null bit, the A/D Converter will respond by shifting out conversion data on each subsequent falling edge of the clock. The most significant bits are clocked out first. The micro is supplying the CS and CLK signals and the A/D Converter responds with the bit data on D_{OUT}.

As shown in Figure 1, starting with an initial NULL bit, bits B11, B10, B9...B0 are shifted out of the A/D Converter. Following bit B0, further CLK falling edges will cause the A/D Converter to shift out bits B1...B11 in reverse order of the initial bit sequence. Continued CLks will shift out zeros following B11 until CS returns high to signal the end of the conversion. On the rising edge of CS, D_{OUT} will change to a Hi-Z state. The device receiving the data from the A/D Converter can use the low-to-high edge of CLK to validate (or latch) the A/D Converter bit data at D_{OUT}.

The 8051 instruction set provides for bit manipulation to allow the use of I/O pins to serve as a serial host for the A/D Converter. By manually toggling the I/O pins and reading the resulting A/D Converter D_{OUT} bits, the designer is free to use any I/O pin that can provide the needed function. The drawback to this method is the bandwidth limit imposed by the execution time of the opcodes supporting the A/D Converter communication. Example 1 shows a code module for a simple I/O port "bit-banging" method for supporting the MCP3201. To optimize for speed, the result is right justified in the ADRESH:ADRESL register pair.

2

Application Notes



SPI is a trademark of Motorola

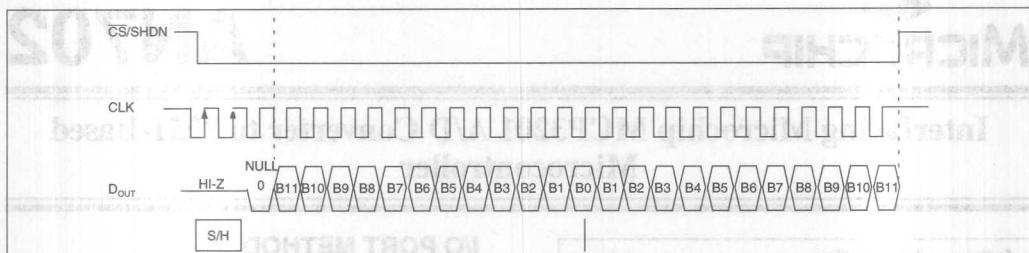


FIGURE 1: MCP3201 Serial Data Format.

EXAMPLE 1: I/O PORT METHOD CODE

```

; GET_AD:    SETB CS      ; set cs hi
             MOV COUNTA,#15      ; init count
             BNE COUNTA, #15, GETAD
; NXTBIT:   CLR DCLK     ; X,X,NULL,D11,D10,D9...D0
             CLR CS      ; CS low to start conversion or keep low till done
             SETB DCLK     ; raise the clock
             MOV C,SDAT   ; put data into C flag
             RLC A        ; shift C into Acc (A/D low bits)
             XCH A,ADRESH ; get ADRESH byte (save low bits in ADRESH for now)
             RLC A        ; shift C into Acc (A.D high bits)
             XCH A,ADRESH ; get low bits back into Acc for next loop
             DJNZ COUNTA,NXTBIT
             MOV ADRESL,A
             ANL ADRESH,#0FH
             SETB CS      ; set CS hi to end conversion

```

USING THE SERIAL PORT IN SYNCHRONOUS MODE0

The UART on the 8051 supports a synchronous shift register mode that, with some software help, can be used to speed up the communications to the A/D Converter. In Mode0, the UART uses the RX pin for data I/O, while the TX pin provides a synchronization clock. The shift register is 8 bits wide and the TX pin will transition low to high to supply a clock rising edge for each bit. Figure 2 shows the typical Mode0 timing.

Since the UART was designed primarily to support RS-232 data transfers, the bit order expected is LSB first. The shift register Mode0 also uses this bit order. As shown in Figure 1, the first 12 bits of the A/D Converter data are ‘backwards’ for our application. Fortunately, the MCP3201 provides the reverse order of sampled bits after the initial transfer of bits B11...B0.

Inspection of Figure 1 readily shows that working back from the last data bit transferred, 3 bytes received from the shift register will cover 24 bits of the 26 bits transferred from the A/D Converter. Conveniently, bit manipulation can be used to provide the two CLK rising edges needed during the beginning sample operation. After these two initial CLK cycles, the UART shifter can be accessed three times to read in the remainder of the data. The bit order will be correct for the third shifter byte as MSB data, the second byte will have 4 LSBs in the upper nibble (the lower nibble will be masked off), and the first byte will be tossed. Figure 3 shows the relationship between the shifted bits and SBUF data received by the UART. Example 2 shows a code module for using the synchronous port as the interface. The result is left justified in the ADRESH:ADRESL register pair.

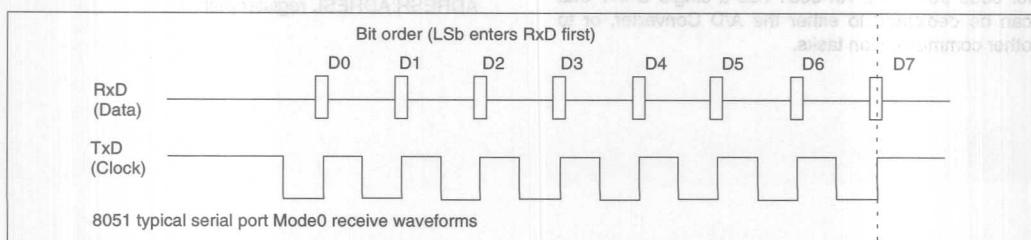


FIGURE 2: Typical 8051 UART Mode0 Timing.

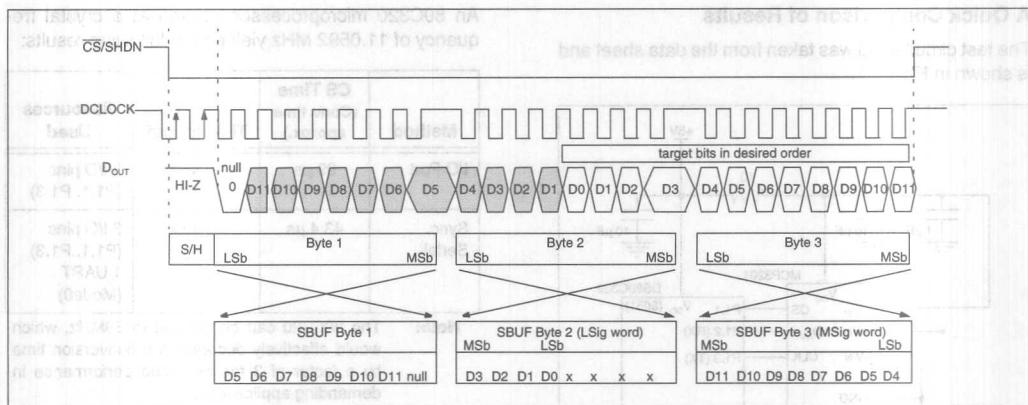


FIGURE 3: Serial Port Waveforms.

EXAMPLE 2: SYNCHRONOUS PORT CODE

```

GET_AD:   SETB  CS      ; set CS hi
          CLR   DCLK    ; X,X,NULL,D11,D10,D9...D0
          CLR   CS      ; CS low to start conversion or keep low till done
          SETB  DCLK    ; 1st S/H clock
          CLR   DCLK    ; 2nd S/H clock and leave DCLK high
          SETB  DCLK    ; 2nd S/H clock and leave DCLK high
          SETB  REN_1   ; REN=1 & R1_1=0 initiates a receive
          CLR   R1_1    ;
BYTE_1:   JNB   R1_1,BYTE_1
          MOV   A,SBUF1  ; toss this byte
          CLR   R1_1    ;
BYTE_2:   JNB   R1_1,BYTE_2
          MOV   ADRESL,SBUF1 ; save LSbs
          CLR   R1_1    ;
BYTE_3:   JNB   R1_1,BYTE_3
          MOV   ADRESH,SBUF1 ; save MSbs
          SETB  CS      ; set CS hi to end conversion
          ANL   ADRESL,#0FH ; mask off unwanted LSB bits

```

A Quick Comparison of Results

The test circuit used was taken from the data sheet and is shown in Figure 4.

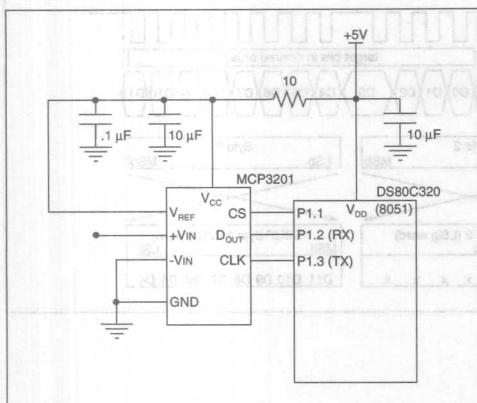


FIGURE 4: Test Circuit.

Oscilloscope screen shots of the I/O port method vs. the Synchronous Port method are shown in Figure 5 and Figure 6.

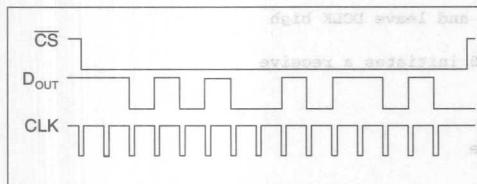


FIGURE 5: Scope Shot: I/O Port Method.

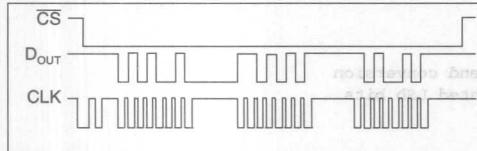


FIGURE 6: Scope Shot: Synchronous Port Method.

An 80C320 microprocessor clocked at a crystal frequency of 11.0592 MHz yielded the following results:

Method	CS Time (Conv. time approx.)	Approx. Throughput	Resources Used
I/O Port	99 µs	10 kHz	3 I/O pins (P1.1..P1.3)
Sync. Serial	43.4 µs	23 kHz	3 I/O pins (P1.1..P1.3) 1 UART (Mode0)

Note: The 80C320 can be clocked to 33MHz, which would effectively decrease the conversion time by a factor of 3 for increased performance in demanding applications.

TABLE 1: Conversion Time Comparison.

IN SUMMARY

Both methods illustrate the ease with which the MCP3201 A/D Converter can complement a design to add functionality for processing analog signals. The synchronous serial port method provides a 2:1 performance increase over the I/O port method, but consumes one UART as a resource. The I/O port method is flexible in allowing any suitable 3 I/O pins to be used in the interface.

Potential applications include control voltage monitoring, data logging, and audio processing. The routines in the source code appendices provide the designer with an effective resource to implement the design.

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX A: I/O PORT SOURCE CODE

```

; Author Lee Studley
; Assembled with Metalink's FreeWare ASM51 assembler
; Tested with NOICE emulation software.
; Tested with a DALLAS DS80C320 (8031) micro clocked @ 11.0592mhz
; This test uses a 'bit banging' approach yielding a conversion time
; of approximately 99uS
; The result is transmitted via the original 8051 UART to an ascii
; terminal at 19.2k baud 8N1 format
;
; ====== RESET AND INTERRUPT VECTORS ======
;
RSTVEC EQU 0000H;
IE0VEC EQU 0003H;
TF0VEC EQU 000BH;
IE1VEC EQU 0013H;
TF1VEC EQU 001BH;
RITIVEC EQU 0023;
TF2VEC EQU 002BH; ( 8052 )
;
; ====== VARIABLES ======
DSEG
;
; ====== PROGRAM VARIABLES ======
COUNTA EQU 30H
COUNTB EQU 31H
ADRESL EQU 2
ADRESH EQU 3
;
; ====== HARDWARE EQUATES ======
DCLK EQU P1.3
SDAT EQU P1.2
CS EQU P1.1
;
; ====== CONSTANTS ======
;
; ====== PROGRAM CODE ======
;
CSEG
;
;org RSTVEC
;LJMP START
;
ORG 4000H ; NOICE SRAM/PROGRAM SPACE
;
;=====
START:
; =====
; Initialize the on-chip serial port for mode 1
; Set timer 1 for baud rate: auto reload timer is quad
; =====
SETUPUART:
        MOV    PCON, #80H; SET FOR DOUBLE BAUD RATE
        MOV    TMOD, #00100010B; two 8-bit auto-reload counters
;
```

```

MOV TH1, #0FDH; 19.2K @ 11.059 MHZ
MOV SCON,#01010010B; mode 1, TI set
SETB TR1; start timer for serial port
=====
; GET_AD: Initiates the A/D conversion and retrieves the AD sample into
; ADRESH,ADRESL.
; The A/D converter is connected to port1 pins 0..2 as:
; SDAT EQU P1.0 I/O
; DCLK EQU P1.1 I/O
; CS EQU P1.2 I/O
; Uses: ADRESL,ADRESH,ACC,COUNTA
; Exits: ADRESH=(x,x,x,x,B11..B8), ADRESL(B7..B0,,)
=====
GET_AD: SETB CS ; set cs hi
MOV COUNTA,#15 ; number of bits to shift 12+X,X,NULL=15
NXTBIT: CLR DCLK ; CS low to start conversion or keep low till done
CLR CS ; raise the clock
SETB DCLK ; put data into C flag
MOV C,SDAT ; shift C into Acc (A/D low bits)
RLC A ; get ADRESH byte(sav low bits in ADRESH for now)
RLC A ; shift C into Acc (A/D high bits)
XCH A,ADRESH ; get low bits back into Acc for next loop
DJNZ COUNTA,NXTBIT
MOV ADRESL,A ; put A into ADRESL
ANL ADRESH,#0FH ; mask off unwanted bits (x,X,X,Null)
SETB CS ; set CS hi to end conversion
;=END__GET_AD=====

;=====
PRODIGS:
CALL BIN16BCD
MOV R0,#7
NXTDIG:
MOV A,#30H
ADD A,@R0
CALL SENDCHAR
DEC R0
CJNE R0,#3,NXTDIG

CALL RETNEWLINE ; send a carriage return and line feed
CALL DELAY1 ; wait here awhile
JMP START

;=====
;=SUBROUTINES=====
;=====
;=====
RETNEWLINE:
MOV A,#0AH ; *** \n newline
CALL SENDCHAR
MOV A,#0DH ; *** return
CALL SENDCHAR
RET

;=====
SENDCHAR:
T_TST: JNB TI,T_TST ; loop till output complete (sat band tol 1)
CLR TI ; clear bit
MOV SBUF,A ; send data
RET
;=====
```

```

;*****
; BIN16BCD
;
; The following routine converts an unsigned integer value in the
; range of 0 - 9999 to an unpacked Binary Coded Decimal number. No
; range checking is performed.
;
; INPUT: R3 (MSB), R2(LSB) contain the binary number to be
; converted.
; OUTPUT: R7(MSD), R6, R5, R4(LSD) contain the 4 digit, unpacked BCD
; representation of the number.
; Uses: R1,R2,R3,R4,R5,R6,R7,ACC
;*****
```

BIN16BCD:

```

MOV R1,#16D      ; loop once for each bit (2 bytes worth)
MOV R5,#0        ; clear regs.
MOV R6,#0
MOV R7,#0
```

BCD_16LP:

```

MOV A,R2
ADD A,R2
MOV R2,A
```

```

MOV A,R3
ADDC A,R3
MOV R3,A
```

```

=====
```

```

MOV A,R5
ADDC A,R5
DA A
MOV R5,A
```

```

MOV A,R6
ADDC A,R6
DA A
MOV R6,A
DJNZ R1,BCD_16LP ; loop until all 16 bits done
```

```

=====
```

;unpack the digits

```

=====
```

```

SWAP A           ;swap so that digit 4 is rightmost
ANL A,#0FH       ;mask off digit 3
MOV R7,A         ;save digit 4 in R7
MOV A,R6         ;get digits 3,4 again
ANL A,#0FH       ;mask off digit 4
MOV R6,A         ;save digit 3
```

```

MOV A,R5           ;get digits 1,2
SWAP A           ;swap so that digit 2 is rightmost
ANL A,#0FH       ;mask off digit 1
XCH A,R5         ;put digit 2 in R5, digit 1 => ACC
ANL A,#0FH       ;mask off digit 2
MOV R4,A         ;save digit 1 in R4 then exit
```

```

RET
```

```

=====
```

```

DELAY1: DJNZ R2,DELAY1
DELAY2: DJNZ R3,DELAY1
RET
=====
END
```

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX B: SYNCHRONOUS PORT SOURCE CODE

```
;  
;  
$MOD51  
$TITLE(ads2)  
$DATE(7/29/98)  
$PAGewidth(132)  
$OBJECT(C:\ASM51\ads2.OBJ)  
;  
; Author: Lee Studley  
; Assembled with Metalink's FreeWare ASM51 assembler  
; Tested with NOICE emulation software.  
; Tested with a DALLAS DS80C320 (8031) micro clocked @ 11.0592mhz  
; This micro has a 2nd UART resource at pins P1.2,P1.3  
;  
; This test uses a the UART MODE0 approach yielding a conversion  
; time of approximately 43.4us  
; The result is transmitted via the original 8051 UART to an ascii  
; terminal at 19.2k baud 8N1 format  
;  
;===== RESET AND INTERRUPT VECTORS =====  
;  
RSTVEC EQU 0000H ;  
IE0VEC EQU 0003H ;  
TFOVEC EQU 000BH ;  
IE1VEC EQU 0013H ;  
TF1VEC EQU 001BH ;  
RITIVEC EQU 0023H ;  
TF2VEC EQU 002BH ; ( 8052 )  
;  
;===== VARIABLES =====  
DSEG  
;  
;===== PROGRAM VARIABLES =====  
COUNTA EQU 30H ;  
COUNTB EQU 31H ;  
ADRESL EQU 2 ;  
ADRESH EQU 3 ;  
;  
;===== HARDWARE EQUATES =====  
DCLK EQU P1.3 ;  
SDAT EQU P1.2 ;  
CS EQU P1.1 ;  
;  
;  
;2nd Uart equates  
SCON1 EQU 0COH ;  
SBUF1 EQU 0C1H ;  
REN_1 BIT SCON1.4 ;  
R1_1 BIT SCON1.0 ;  
;  
;  
;===== CONSTANTS =====  
;  
;===== PROGRAM CODE =====  
;  
CSEG  
;  
ORG RSTVEC  
LJMP START  
ORG 4000H ; NOICE SRAM/PROGRAM SPACE
```

```

=====
START:
=====
; Initialize the on-chip serial port for mode 1
; Set timer 1 for baud rate: auto reload timer
=====
SETUPUART:
    MOV PCON,#80H          ; SET FOR DOUBLE BAUD RATE
    MOV TMOD,#00100010B     ; two 8-bit auto-reload counters
    MOV TH1,#0FDH           ; 19.2K @ 11.059 MHZ
    MOV SCON,#01010010B      ; mode 1, TI set
    SETB TR1                ; start timer for serial port
=====
SETUPUART2:
    MOV SCON1,#00000000B    ; 2nd uart mode 0, TI set
                           ; Shift clk(TX)=Tosc/12
=====
; =====
; GET_AD: Initiates the A/D conversion and retrieves the AD sample into
; ADRESH,ADRESL.
; The A/D converter is connected to port1 pins 1..3 as:
; DCLK    EQU P1.3 Tx(synchronous clock)
; SDAT    EQU P1.2 Rx(synchronous data)
; CS      EQU P1.1 I/O
; Uses: ADRESL,ADRESH,ACC,COUNTA
; Exits: ADRESH=(B11..B4), ADRESL(B3..B0,x,x,x,x)
; =====
GET_AD:   SETB CS          ; set CS hi
           CLR DCLK         ; X,X,NULL,D11,D10,D9...D0
           CLR CS           ; CS low to start conversion or keep low till done
           SETB DCLK         ; 1st S/H clock
           CLR DCLK         ;
           SETB DCLK         ; 2nd S/H clock and leave DCLK high
;
           SETB REN_1        ; REN=1 & R1_1=0 initiates a receive
           CLR R1_1          ; end without select
;
BYTE_1:   JNB R1_1,BYTE_1
           MOV A,SBUF1       ; toss this byte
           CLR R1_1
;
BYTE_2:   JNB R1_1,BYTE_2
           MOV ADRESL,SBUF1  ; save lsbs
           CLR R1_1
;
BYTE_3:   JNB R1_1,BYTE_3
           MOV ADRESH,SBUF1  ; save msbs
           SETB CS           ; set CS hi to end conversion
           ANL ADRESL,#0F0H   ; mask off unwanted lsb bits
;
;=END__GET_AD=====
; =====
PROCDIGS:
    CALL BIN16BCD
    MOV R0,#7
;
NXTDIG:
    MOV A,#30H
    ADD A,@R0
    CALL SENDCHAR
    DEC R0
    CJNE R0,#3,NXTDIG
;
    CALL RETNEWLINE      ; send a carriage return and line feed
;

```



```
ADDC A,R5
DA A
MOV R5,A

MOV A,R6
ADDC A,R6
DA A
MOV R6,A
DJNZ R1,BCD_16LP ; loop until all 16 bits done
;=====
;unpack the digits
;=====

SWAP A ;swap so that digit 4 is rightmost
ANL A,#0FH ;mask off digit 3
MOV R7,A ;save digit 4 in R7
MOV A,R6 ;get digits 3,4 again
ANL A,#0FH ;mask off digit 4
MOV R6,A ;save digit 3

MOV A,R5 ;get digits 1,2
SWAP A ;swap so that digit 2 is rightmost
ANL A,#0FH ;mask off digit 1
XCH A,R5 ;put digit 2 in R5, digit 1 => ACC
ANL A,#0FH ;mask off digit 2
MOV R4,A ;save digit 1 in R4 then exit

RET

;=====
;=====
;=====

DELAY1: DJNZ R2,DELAY1
DELAY2: DJNZ R3,DELAY1
RET
;=====
;=====
;=====

END
```

AN702

NOTES:

Using the MCP320X 12-Bit Serial A/D Converter with Microchip PICmicro® Devices

Author: Jake McKernan
Microchip Technology Inc.

OVERVIEW

The MCP320X devices comprise a family of 12-bit successive approximation Analog to Digital (A/D) Converters. These devices provide from one to eight analog inputs with both single ended and differential inputs. Data is transferred to and from the MCP320X through a simple SPI™-compatible 3-wire interface. This application note discusses how to interface the MCP320X devices to Microchip PICmicro® devices, using both software and hardware SPI with examples shown in C and Assembly languages. The programs in this application note were developed using a PIC16C62A and MCP3202 on a PICDEM-2 demonstration board. As a matter of convenience, the CLK, D0, and Di pins of the PIC16C62A are used for all examples, whether using the hardware SPI peripheral or the software SPI implementation. The software SPI may be adapted to I/O ports on any PICmicro device.

COMMUNICATION

Communication to the MCP3202 is accomplished via a synchronous SPI-compatible scheme. This interface consists of three lines; DOUT, DIN and CLK. Control information is loaded into the MCP320X through the DIN line and data is output on the DOUT line. The CLK signal is generated by the PICmicro and is used as both communication and conversion clock for the A/D Converter. Data bits are latched in from DIN on the rising edge of CLK and latched out to DOUT on the falling edge. A fourth line, CS, is an active low signal used to select the chip and enable it for conversion and communication. See Figure 1 for a communication timing diagram.

2
Application
Notes

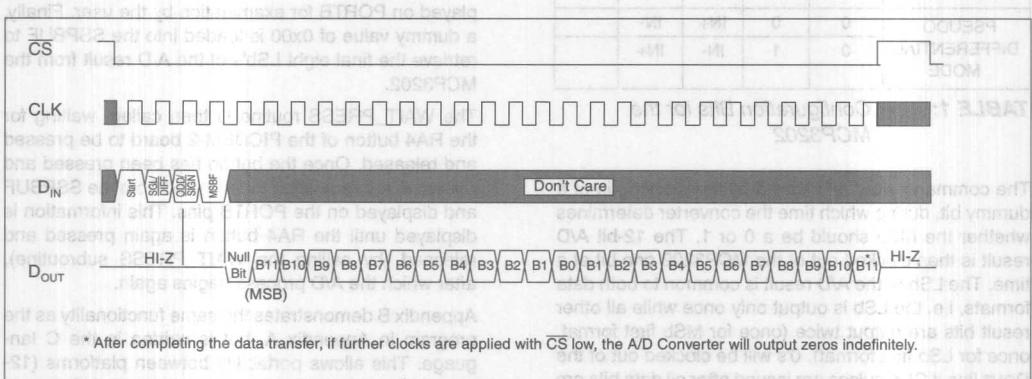


FIGURE 1: Communication with MCP3202 using LSB first format

After completing the data transfer, if further clocks are applied with CS low, the A/D Converter will output zeros indefinitely.

CS This signal is asserted low to enable the MCP3202. It must be asserted for at least one clock cycle after the last bit has been shifted in. If CS is asserted low for longer than one clock cycle, the A/D converter will continue to output zeros indefinitely.

CLK This signal is the clock for the A/D converter. It must be a valid digital signal with a minimum frequency of 100 ns. It is generated by the PICmicro and is used as both the communication clock and the conversion clock for the A/D converter.

DIN This signal is the data input to the A/D converter. It is latched on the rising edge of the clock. The data is shifted in from MSB to LSB. The data is valid for one clock cycle.

DOUT This signal is the data output from the A/D converter. It is shifted out from LSB to MSB. The data is valid for one clock cycle.

A 4-bit configuration command is issued to the MCP3202 to begin the conversion process. When communication of the command word to the MCP3202 begins, the first '1' bit seen by the MCP3202 on the DIN line will be interpreted as a start bit. Leading 0's may be clocked into the device with no effect. The start bit is followed by a mode selection bit, indicating whether the conversion result will be single-ended or differential. A mode select bit of '1' selects single-ended mode and '0' selects differential mode. Next, the channel select bit is clocked into the MCP3202, which sets the channel to be converted. A '0' in this bit position selects Channel 0, while a '1' selects Channel 1. If differential mode was selected, the channel select bit determines which channel will be subtracted from the other. Table 1 illustrates how the A/D result will be affected by the channel and mode selection bits. Finally, a data format bit is clocked into the MCP3202. This bit selects whether the result of the conversion will be shifted out in LSb format. A '0' in this bit position will cause the data to be shifted out in MSb only format. If a '1', the data will first be shifted out in MSb format, followed by the same data in LSb format. Keep in mind that the data will *always* be shifted out in MSb format, regardless of the state of the data format bit.

	CONFIG BITS		CHANNEL SELECTION		GND
	SGL/ DIFF	ODD/ SIGN	0	1	
SINGLE ENDED MODE	1	0	+		-
	1	1		+	-
PSEUDO- DIFFERENTIAL MODE	0	0	IN+	IN-	
	0	1	IN-	IN+	

TABLE 1: Configuration Bits for the MCP3202

The command word is followed by the clocking in of a dummy bit, during which time the converter determines whether the MSb should be a 0 or 1. The 12-bit A/D result is then clocked out of the MCP3202 one bit at a time. The LSb of the A/D result is common to both data formats, i.e. the LSb is output only once while all other result bits are output twice (once for MSb first format, once for LSb first format). 0's will be clocked out of the DOUT line if CLK pulses are issued after all data bits are extracted from the converter.

IMPLEMENTATION

As previously mentioned, several code examples of interfacing to the MCP3202 are shown in this application note. All methods use essentially the same algorithm of performing an A/D conversion, displaying the result on PORTB, then waiting for a keypress. The examples cover hardware and software SPI, relocatable and absolute assembly and C.

Written in absolute assembly, Appendix A shows the use of the hardware SSP module in master SPI mode. The SSP is set up to clock data in on the rising edge, clock data out on the falling edge and drive the clock high when idle, with a frequency of Fosc/64. All bits of PORTB are configured as outputs and the port is cleared. To begin the conversion process, the MCP3202 is selected using the CS line and 0x01 is loaded into the SSPBUF of the PIC16C62A. This shifts out seven leading 0's, followed by a start bit. The subroutine WAIT_BF then monitors the BF flag in the SSPSTAT register, which indicates when the 8-bit transfer is complete. Next, a value of 0xE0 is loaded into the SSPBUF, the MSb's being the three configuration information bits, and the lower five bits being dummy information to round out the byte. The configuration bits in this example set the MCP3202 up for single-ended conversion on channel 1, with the output in MSb first format. During the transmission of the 5 LSb's, the MCP3202 will begin shifting out A/D result data. The WAIT_BF subroutine is called after the SSPBUF is loaded, waiting for the transmission to be complete. Once the transmission is complete, the MSb's of the result are read from the SSPBUF, masked, and displayed on PORTB for examination by the user. Finally, a dummy value of 0x00 is loaded into the SSPBUF to retrieve the final eight LSb's of the A/D result from the MCP3202.

The WAIT_PRESS routine is then called, waiting for the RA4 button of the PICDEM-2 board to be pressed and released. Once the button has been pressed and released, the remaining data is read from the SSPBUF and displayed on the PORTB pins. This information is displayed until the RA4 button is again pressed and released (by calling the WAIT_PRESS subroutine), after which the A/D process begins again.

Appendix B demonstrates the same functionality as the program in Appendix A, but is written in the C language. This allows portability between platforms (12-bit, 14-bit or 16-bit cores), with a minimum of change to the program.

Appendices C and D are used together to show a hardware SPI implementation using relocatable assembly code. The main file (MCP3202c.asm) is shown in Appendix C and contains the main functionality of the program, while the assembly file shown in Appendix D (waitfcn.asm) contains the auxiliary functions (i.e. waiting for SPI transmission to complete and for RA4 press and release). The linker script (16c62a.lkr) shown in

Appendix D controls where the relocatable segments are placed in the PIC16C62A program memory and defines the processor's available RAM space for the linker. Please consult the MPASM User's Guide for more details on how to write relocatable code.

Appendix E illustrates communication to the MCP3202 using firmware SPI rather than the hardware peripheral. The same I/O pins are used to generate the clock and data signals as with the hardware peripheral, for convenience. Program initialization occurs as with the previous examples, except that the hardware peripheral is excluded and replaced with initialization of PORTC bits. Three registers are initialized to be used as input and output buffers, and there are two new subroutines added to communicate to the MCP3202. The first routine called will be OUT_CONTROL, which issues the control word to the MCP3202. The control word to be sent is loaded into the OUTBUF register before the subroutine is called. Each of the four bits is then shifted out and clocked into the A/D Converter using the DOUT and CLK lines of PORTC, respectively. Once all bits are shifted out, the subroutine returns to the calling function. To retrieve the data from the A/D Converter, a second subroutine is implemented. The IN_DATA subroutine toggles the CLK line and reads the DIN line, shifting each new bit into the INBUFL and INBUFH registers. All 12 bits of the result are read by this subroutine which will return to the calling function once the transfer is complete. As with the previous examples, the MSb's are displayed on PORTB, while the program waits for RA4 to toggle. The LSb's are then displayed, the program waits for RA4 to toggle again, and the process repeats again.

Appendix F is a variation on Appendix E, demonstrating the use of relocatable assembly to implement a software SPI. The same subroutines are used for this example, but are declared as external. The wait functions and linker script (waiffcn.asm, 16c62a.lkr) files shown in Appendix C are used in this example. The ser_io.asm file shown in Appendix G contains the OUT_CONTROL and IN_DATA subroutines used in this example.

The final example, shown in Appendix H, illustrates the firmware SPI implementation in the C language. Two functions are added to this implementation, Output_Control and Input_Data. As with the previous example, the Output_Control shifts the 4-bit command out to the MCP3202 one bit at a time and Input_Data reads all 12 bits of the result. The data is then displayed on PORTB, waiting for input on RA4 before continuing on. In this program, the A/D result data may be accessed in one of two ways; as a 16-bit value or as two 8-bit values. When reading the value in from the MCP3202 using the Input_Data function, the A/D result is treated as a 16-bit value. During the display portion of the program, the result is accessed 8-bits at a time for display on PORTB.

SCHEMATIC

The code for this application note was developed on a PICDEM-2 demonstration board. An equivalent circuit of the board as used in this application note is shown in Appendix I. A full schematic of the PICDEM-2 board can be found in the PICDEM-2 User's Guide, available with the kit or from the Microchip web site (www.microchip.com).

The SPI communication lines CLK, DOUT and DIN are connected to RC3, RC4 and RC5, respectively. The CS signal is generated using RC2 as a general purpose output pin. PORTB is used entirely as an output port for display of A/D result data. All LED's are driven through 470Ω current limiting resistors. RA4 is connected to a momentary contact switch and pullup resistor for allowing the user to cycle through the A/D result data on PORTB.

Channel 1 of the A/D Converter is used throughout the application note, and must have an analog voltage applied to it to get meaningful results from the MCP3202. This was done using a 0.5v power supply output fed directly into pin three of the MCP3202.

The PIC16C62A uses the RC oscillator configuration as the main clock, operating at an approximate frequency of 4MHz. An RC network is also provided on the MCLR line to help ensure that the device is reset correctly on application of power.

CONCLUSION

The example code shown in this application note gives a firm grasp of how to interface the MCP3202 A/D Converter to PICmicro devices. The code has the potential to be adapted to any Microchip PICmicro device, an exercise left up to the user. Implementations in multiple languages and styles also gives the developer flexibility in successfully writing code and libraries to use this device in end-user applications.

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX A: HARDWARE SPI, ABSOLUTE ASSEMBLY

```

;***** This program demonstrates communication with the MCP3202 A/D converter
; This code was written for the midrange PICmicro devices (using a PICDEM-2 board and the 16C62A) and uses the SSP module in SPI mode for communication to the MCP3202.
; Filename: mcp3202a.asm
; (C), 1998 Microchip Technology, Inc.
; All Rights Reserved

list p=16c62a

include "p16c62a.inc"

ADCS equ 0x02 ;chip select line for A/D

ORG 0x0000
clrf PCLATH ;reset PCLATH for Page0 operation
clrf STATUS ;reset STATUS for Bank 0 operation
clrf FSR ;clear FSR
goto START ;begin main program

ORG 0x0004
_ISR
goto _ISR ;stay here if interrupt occurs
WAIT_BF
bsf STATUS,RP0 ;select Bank0
btfs SSPSTAT,BF ;check for BF set
goto WAIT_BF ;continue to wait
bcf STATUS,RP0 ;select Bank1
return ;return to caller

WAIT_PRESS
btfsc PORTA,4 ;check for button press
goto WAIT_PRESS

WAIT_RLS
btfss PORTA,4 ;check for button release
goto WAIT_RLS
return ;return to caller

START
movlw 0x32 ;set up SSP to clock data out on falling edge
movwf SSPCON ;clock data in on rising edge, clock idle high
clr PORTB ;clear PortB outputs

```

```

bsf STATUS,RP0           ;select Bank1
movlw 0x10
movwf TRISC             ;set up Port C for SPI master

clrf TRISB              ;configure PortB as outputs

bcf STATUS,RP0           ;select Bank0
bsf PORTC,ADCS          ;deselect A/D device

BEGIN_AD
    bcf PORTC,ADCS      ;select A/D device
    movlw 0x01
    movwf SSPBUF           ;output start bit

    call WAIT_BF          ;wait for transfer complete

    movlw 0xE0
    movwf SSPBUF           ;output 3 command and 5 dummy bits
    call WAIT_BF          ;shift out command and receive 4 MSb's
                           ;wait for transfer complete

    movf SSPBUF,W          ;read result (MSb's of conversion)
    andlw 0x0F
    movwf PORTB             ;mask out MSb's
                           ;display on PortB

    movlw 0x00
    movwf SSPBUF           ;shift remaining bits
    call WAIT_BF          ;wait for transfer complete

    call WAIT_PRESS         ;wait for button press/release before advancing

    movf SSPBUF,W          ;read result (LSb's)
    movwf PORTB             ;display on PortB

    bsf PORTC,ADCS          ;de-select A/D converter
    call WAIT_PRESS         ;wait for button press/release before advancing

HERE
    goto BEGIN_AD          ;play it again, Sam
                           ;wasn't scared for me

END

```

APPENDIX B: HARDWARE SPI, C LANGUAGE

```
/*
 * This program is written to demonstrate interfacing the MCP3202 A/D converter to Microchip PICmicro devices. The code demonstrates how to implement hardware SPI to communicate with the converter, and is written in C for the HiTech PICC C compiler. By modifying the #include statement to "#include<16c62a.h>" the code may be compiled using MPLAB-C 1.21.
 *
 * Filename: mcp3202b.c
 *
 * (C) 1998 Microchip Technology, Inc.
 * All Rights Reserved
 *
 ****
 */
#include<pic1662.h>      /* modify this statement for use with the MPLAB-C compiler */

#define ADCS 0x04          /* I/O bit position for CS line */
#define BUSY 0x01           /* Bit0 of SSPSTAT, indicated when SPI xmission complete */
#define BUTTON 0x10          /* I/O bit position for RA4 line */

void Wait_for_Press()
{
    while(PORTA & BUTTON)
    {
        /* wait for button press */ /* a D/A converter */
    }
    while(!(PORTA & BUTTON))
    {
        /* wait for button release */ /* message at value */
    }
}

void main(void)
{
    TRISB = 0x00;
    PORTB = 0x00;           /* reset PortB outputs */

    SSPCON = 0x32;          /* set up SSP to clock data out on falling edge */
    TRISC = 0x10;           /* clock data in on rising edge, clock idle high */

    PORTC |= ADCS;          /* de-select A/D device */

    while(1)
    {
        PORTC &= ~ADCS;     /* select A/D device */
        SSPBUF = 0x01;       /* output start bit */

        while(!(SSPSTAT & BUSY))
```


Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX C: HARDWARE SPI, RELOCATABLE ASSEMBLY

```
;*****
;*
;*      This program demonstrates communication with the MCP3202 A/D converter
;*      using relocatable assembly code.  This code was written for the midrange
;*      PICmicro devices (using a PICDEM-2 board and the 16C62A) and uses the SSP
;*      module in SPI mode for communication to the MCP3202.
;*
;*      The two subroutines WAIT_BF and WAIT_PRESS are external functions, compiled
;*      and linked separately from the WAITFCN.ASM file.  These subroutines wait
;*      for the SPI transmission to complete and for RA4 to be pushed and released,
;*      respectively.
;*
;*      Filename: mcp3202c.asm
;*
;*      (C) 1998 Microchip Technology, Inc.
;*      All Rights Reserved
;*
;*****
```

```
list p=16C62a
#include "p16c62a.inc"
ADCSequ0x02          ;CS line for MCP3202 (RC6)

EXTERN WAIT_BF         ;define wait function call symbols
EXTERN WAIT_PRESS

RESETCODE              ;select reset code section
    clrf PCLATH        ;reset PCLATH on powerup
    clrf STATUS         ;reset STATUS on powerup
    clrf FSR            ;reset FSR on powerup
    goto START          ;go start and initialize program

INTCODE                ;select interrupt code section
    _ISR
    goto _ISR           ;stay here if interrupt occurs

START                  ;initialization
    movlw 0x32          ;setup SSP for operation
    movwf SSPCON

    clrf PORTB          ;reset LED output port

    bsf STATUS,RP0       ;select Bank1
    movlw 0x10
    movwf TRISC          ;configure PORTC for operation

    clrf TRISB          ;configure PORTB as outputs

    bcf STATUS,RP0       ;select Bank0
    bsf PORTC,ADCS      ;deselect A/D converter
```

```

BEGIN_AD          ;start A/D conversion
bcf PORTC,ADCS   ;select A/D converter
movlw 0x01         ;load start bit
movwf SSPBUF      ;output start bit to A/D

call WAIT_BF      ;wait for transmission complete

movlw 0xE0         ;load 3 command and 5 dummy bits
movwf SSPBUF      ;output on SPI port

call WAIT_BF      ;wait for transmission complete

movf SSPBUF,W     ;read A/D result MSb's
andlw 0x0F        ;mask off garbage bits
movwf PORTB       ;output MSb's on PORTB LED's

movlw 0x00         ;load dummy data
movwf SSPBUF      ;output on SPI (shifts in Lsb's)
call WAIT_BF      ;wait for transmission complete

call WAIT_PRESS    ;wait for button press/release

movf SSPBUF,W     ;read A/D result Lsb's
movwf PORTB       ;output Lsb's on PORTB LED's

bsf PORTC,ADCS    ;deselect A/D converter

call WAIT_PRESS    ;wait for button press/release

HERE             ;repeat process
goto BEGIN_AD

END

```

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX D: WAIT FUNCTIONS AND LINKER SCRIPT FOR APPENDIX C

```
;*****
;*
;*      Wait functions for MCP3202 A/D converter demonstration. These
;*      functions wait for SPI communication and RA4 button press/release
;*      on the PICDEM-2 board. This file is to be assembled and linked
;*      with mcp3202c.ASM or mcp3202e.ASM for proper usage.
;*
;*      Filename: waitfcn.asm
;*
;*      (C) 1998 Microchip Technology, Inc.
;*      All Rights Reserved
;*
;*****
```

```
list p=16C62a
#include "p16c62a.inc"
CODE
```

```
WAIT_BF
GLOBAL WAIT_BF
bsf STATUS,RPO
btfss SSPSTAT,BF
goto WAIT_BF
bcf STATUS,RPO
return
```

```
;wait for SPI transmission complete
;declare WAIT_BF visible to outside world
;select Bank1
;check for transmission complete (BF set)
;not finished, continue waiting
;select Bank0
;return to calling function
```

```
WAIT_PRESS
GLOBAL WAIT_PRESS
btfs PORTA,4
goto WAIT_PRESS
```

```
;wait for RA4 press/release
;declare WAIT_PRESS visible to outside world
;check for button press
;not pressed, check again
```

```
WAIT_RLS
btff PORTA,4
goto WAIT_RLS
return
```

```
;button now pressed
;check for button release
;not released, check again
;button now released, return to calling func
```

```
END
```

```
/*
**      16C62A Linker Script to be used with MCP3202C.ASM and WAITFCN.ASM
**      to link the corresponding object files.
**
**      Filename: 16c62a.lkr
**
**      (C) 1998 Microchip Technology, Inc.
**      All Right Reserved
*/
CODEPAGE NAME=reset_vector START=0x00 END=0x03
CODEPAGE NAME=interrupt_vector START=0x04 END=0x7FF
DATABANK NAME=gpr0    START=0x20    END=0x7F
DATABANK NAME=gpr1    START=0xA0    END=0xBF
DATABANK NAME=sfr0    START=0x00    END=0x1F    PROTECTED
DATABANK NAME=sfr1    START=0x80    END=0x9F    PROTECTED
SECTION NAME=RESET ROM=reset_vector
SECTION NAME=INT ROM=interrupt_vector
```

APPENDIX E: FIRMWARE SPI, ABSOLUTE ASSEMBLY

```

;*****
;*
;This program demonstrates communication with the MCP3202 A/D converter AT&T_M1
;using absolute assembly code. This code was written for the midrange AT&T_M1
;PICmicro devices (using a PICDEM-2 board and the 16C62A) and uses firmware AT&T_M1
;to implement the SPI module for communication to the MCP3202. AT&T_M1
;*
;Filename: mcp3202d.asm AT&T_M1
;*
;(C) 1998 Microchip Technology, Inc. AT&T_M1
;All Rights Reserved AT&T_M1
;*****
list p=16c62a

include "p16c62a.inc"

ADCS    equ    0x02      ;chip select line for A/D converter
DOUT    equ    0x05      ;serial data out to A/D converter
DIN     equ    0x04      ;serial data in from A/D converter
CLK     equ    0x03      ;serial data clock to A/D converter
CBLOCK  0x20

OUTBUF
INBUFH
INBUFL
COUNT
ENDC

ORG 0x0000
clrf PCLATH      ;reset PCLATH for Page0 operation
clrf STATUS       ;reset STATUS for Bank 0 operation
clrf FSR          ;clear FSR
goto START        ;begin main program

ORG 0x0004
_ISR
goto _ISR         ;stay here if interrupt occurs

OUT_CONTROL
    movwf OUTBUF      ;load control word into buffer
    swapf OUTBUF      ;rotate control word into position
    movlw 0x04          ;init bit counter

BIT_OUT
    rlf OUTBUF        ;rotate bit into carry
    bcf PORTC,DOUT    ;pre-clear data out
    btfsc STATUS,C    ;check if bit should be set
    bsf PORTC,DOUT    ;set data out

    bsf PORTC,CLK      ;generate clock pulse
    nop
    bcf PORTC,CLK


```

```

        decfsz COUNT           ;decrement bit counter
        goto BIT_OUT          ;output next bit
        return                 ;finished, return to caller

;-----[APPENDIX C]-----;

IN_DATA
        clrf INBUFH           ;clear input buffer
        clrf INBUFL           ;reset input buffer
        movlw 0x0D             ;init bit counter
        movwf COUNT

BIT_IN
        bsf PORTC,CLK         ;set clock to latch bit
        bcf STATUS,C           ;pre-clear carry
        btfsc PORTC,DIN       ;check for high or low bit
        bsf STATUS,C           ;set carry bit

        rlf INBUFL            ;rotate bit into position
        rlf INBUFH
        bcf PORTC,CLK         ;drop clock for next bit

        decfsz COUNT           ;decrement bit counter
        goto BIT_IN            ;get next bit
        return                 ;return to caller

WAIT_PRESS
        btfsc PORTA,0x04       ;check for button press
        goto WAIT_PRESS

WAIT_RLS
        btfss PORTA,0x04       ;check for button release
        goto WAIT_RLS          ;return to caller

START
        clrf PORTB             ;clear PortB outputs
        movlw 0x40             ;initialize PortC: ADCS high, DO, CLK low
        movwf PORTC

        bsf STATUS,RPO          ;select Bank1
        movlw 0x10
        movwf TRISC             ;set up Port C for SPI master

        clrf TRISB              ;configure PortB as outputs

        bcf STATUS,RPO          ;select Bank0
        clrf OUTBUF             ;reset output buffer
        clrf INBUFH             ;reset input buffer
        clrf INBUFL

BEGIN_AD
        bcf PORTC,ADCS          ;select A/D converter

```

```

movlw 0x0F          ;load control word
call OUT_CONTROL    ;output control word

call IN_DATA         ;read data from A/D converter
bsf PORTC,ADCS      ;de-select A/D converter

movlw 0x0F          ;load MSB mask
andwf INBUFH,W       ;mask out MSB's and put result in W
movwf PORTB          ;output MSB's

call WAIT_PRESS      ;wait for button press
movf INBUFL,W        ;load LSB's into W
movwf PORTB          ;output LSB's

call WAIT_PRESS      ;wait for button press
goto BEGIN_AD        ;play it again, Sam

END

```

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX F: FIRMWARE SPI, RELOCATABLE ASSEMBLY

```
;*****
;*
;*      This program demonstrates communication with the MCP3202 A/D converter
;*      using relocatable assembly code.  This code was written for the midrange
;*      PICmicro devices (using a PICDEM-2 board and the 16C62A) and uses the SSP
;*      module in SPI mode for communication to the MCP3202.
;*
;*      The subroutine WAIT_PRESS is an external function, compiled and linked
;*      separately from the WAITFCN.ASM file.  This subroutine waits for RA4 to be
;*      pushed and released.
;*      The subroutines OUT_CONTROL and IN_DATA are also external functions, but
;*      compiled and linked from the SER_IO.ASM file.  INBUFH and INBUFL are data
;*      bytes that are used by the IN_DATA routine to return the A/D conversion
;*      result to the calling function.
;*
;*      Filename: mcp3202e.asm
;*
;*      (C) 1998 Microchip Technology, Inc.
;*      All Rights Reserved
;*
;*****
list p=16c62a

include "p16c62a.inc"

EXTERN WAIT_PRESS
EXTERN OUT_CONTROL
EXTERN IN_DATA

EXTERN INBUFH
EXTERN INBUFL

ADCS    equ     0x02          ;chip select line for A/D converter

RESET   CODE
clrfl PCLATH           ;reset PCLATH for Page0 operation
clrfl STATUS            ;reset STATUS for Bank 0 operation
clrfl FSR               ;clear FSR
goto START              ;begin main program

INT     CODE
_ISR
goto _ISR               ;stay here if interrupt occurs

START
clrfl PORTB             ;clear PortB outputs
movlw 0x40
movwf PORTC             ;initialize PortC: ADCS high, DO, CLK low
bsf STATUS,RP0           ;select Bank1
movlw 0x10
movwf TRISC              ;set up Port C for SPI master
```


Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX G: RELOCATABLE ASSEMBLY FIRMWARE SPI FUNCTIONS FOR APPENDIX F

```
;*****  
;  
;*      Serial functions for MCP3202 A/D converter demonstration. These  
;*      functions perform SPI communication. This file is to be assembled  
;*      and linked with mcp3202e.ASM for proper usage.  
;  
;*      Filename: ser_io.asm  
;  
;*      (C) 1998 Microchip Technology, Inc. 80M 80M  
;*      All Rights Reserved  
;  
;*****  
list p=16c62a  
  
#include "p16c62a.inc"  
  
DOUT    equ    0x05          ;serial data out to A/D converter  
DIN     equ    0x04          ;serial data in from A/D converter  
CLK     equ    0x03          ;serial data clock to A/D converter  
  
UDATA 0x20  
OUTBUF res 1  
INBUFH res 1  
INBUFL res 1  
COUNT  res 1  
  
GLOBAL INBUFH  
GLOBAL INBUFL  
  
CODE  
  
OUT_CONTROL  
GLOBAL OUT_CONTROL  
    movwf OUTBUF           ;load control word into buffer  
    rlf OUTBUF  
    rlf OUTBUF  
    rlf OUTBUF  
    rlf OUTBUF           ;rotate control word into position  
  
    movlw 0x04  
    movwf COUNT            ;init bit counter  
  
BIT_OUT  
    rlf OUTBUF           ;rotate bit into carry  
    bcf PORTC,DOUT        ;pre-clear data out  
    btfsc STATUS,C         ;check if bit should be set  
    bsf PORTC,DOUT        ;set data out  
  
    bsf PORTC,CLK          ;generate clock pulse  
    nop  
    bcf PORTC,CLK  
  
    decfsz COUNT           ;decrement bit counter  
    goto BIT_OUT           ;output next bit
```

```

;finished, return to caller

;FIRMWARE_SPL LANGUAGE
;APPLICABILITY

IN_DATA
GLOBAL IN_DATA
    clrf INBUFH ;reset input buffer
    clrf INBUFL ;reset input buffer
    movlw 0x0D ;init bit counter
    movwf COUNT ;to say we're wolf

BIT_IN
    bsf PORTC,CLK ;set clock to latch bit
    bcf STATUS,C ;pre-clear carry
    btfsc PORTC,DIN ;check for high or low bit
    bsf STATUS,C ;set carry bit

    rlf INBUFL ;rotate bit into position

    bcf PORTC,CLK ;drop clock for next bit

    decfsz COUNT ;decrement bit counter
    goto BIT_IN ;get next bit
    return ;return to caller

END

```

APPENDIX H: FIRMWARE SPI, C LANGUAGE

```
*****  
* This program is written to demonstrate interfacing the MCP3202 A/D  
* converter to Microchip PICmicro devices. The code demonstrates  
* how to implement software SPI to communicate with the converter,  
* and is written in C for the HiTech C compiler, PICC. Changing the  
* #include directive to "#include<16c62a.h>" will allow the use of the  
* MPLAB-C v1.21 C compiler to compile this file.  
*  
* Filename: mcp3202f.c  
*  
* (C) 1998 Microchip Technology, Inc.  
* All Rights Reserved  
*  
*****  
#include <pic1662.h>      /* modify this statement for use with the MPLAB-C compiler */  
  
#define ADCS 0x04          /* I/O bit position for CS line */  
#define BUSY 0x01           /* Bit0 of SSPSTAT, indicated when SPI xmission complete */  
#define BUTTON 0x10          /* I/O bit position for RA4 line */  
  
#define DOUT 0x20           /* data out to MCP3202 */  
#define DIN 0x10             /* data in from MCP3202 */  
#define CLK 0x08              /* clock out to MCP3202 */  
  
/* Function Prototypes */  
  
void Wait_for_Press();  
void Output_Control(char TempChar);  
int Input_Data(void);  
  
  
void Wait_for_Press()  
{  
    while(PORTA & BUTTON)  
    {  
        /* wait for button press */  
    }  
  
    while(!(PORTA & BUTTON))  
    {  
        /* wait for button release */  
    }  
}  
  
void Output_Control(char TempChar)  
{  
    unsigned char Mask = 0x08;          /* mask to test for 0/1 */  
    unsigned char Count;                /* gen purpose bit counter */  
  
    for(Count = 0x00; Count < 0x04; Count++) /* count 4 bits */  
}
```

```

{
    PORTC &= ~DOUT;           /* pre-clear data line */          0x<0 = CS1INT
                                /* clear portc before writing */      0x00 = CS1INT
    if(TempChar & Mask)        /* check if bit should be high or low */ 0x01 = CS1INT
    {
        PORTC |= DOUT;        /* set data line */                  0x02 = CS1INT
    }
                                /* read/write C/A clock-low */
    PORTC |= CLK;            /* send clock line high */          0x03 = CS1INT
                                /* rotate mask for next bit */      0x04 = CS1INT
                                /* also used to burn time for clock */ 0x05 = CS1INT
    PORTC &= ~CLK;           /* send clock line low */          0x06 = CS1INT
}
                                /* read/write C/A clock-high */
int Input_Data(void)
{
    unsigned char Count;       /* gen purpose bit counter */      0x07 = CS1INT
    unsigned int Mask = 0x8000; /* mask to insert '1' at bit position */ 0x08 = CS1INT
    unsigned int Result = 0x0000; /* A/D result register */          0x09 = CS1INT
                                /* read/write C/A validbit */
    for(Count = 0x00; Count < 0x0D; Count++) /* count 13 bits */
    {
        if(PORTC & DIN)        /* 12-bit result + 1 null bit */ 0x0A = CS1INT
        {
            Result |= Mask;   /* check if bit is high or low */
                                /* bit high, set bit in result */
        }

        PORTC |= CLK;          /* send clock line high */          0x0B = CS1INT
                                /* rotate mask for next bit */      0x0C = CS1INT
                                /* also used to burn time for clock */ 0x0D = CS1INT
        PORTC &= ~CLK;          /* send clock line low */          0x0E = CS1INT
    }

    Result >>= 0x03;         /* rotate bits into position */     0x0F = CS1INT
    Result &= 0x0FFF;         /* mask out 12-bit result */       0x10 = CS1INT
                                /* read/write C/A validbit */

    return(Result);          /* return result to caller */       0x11 = CS1INT
}

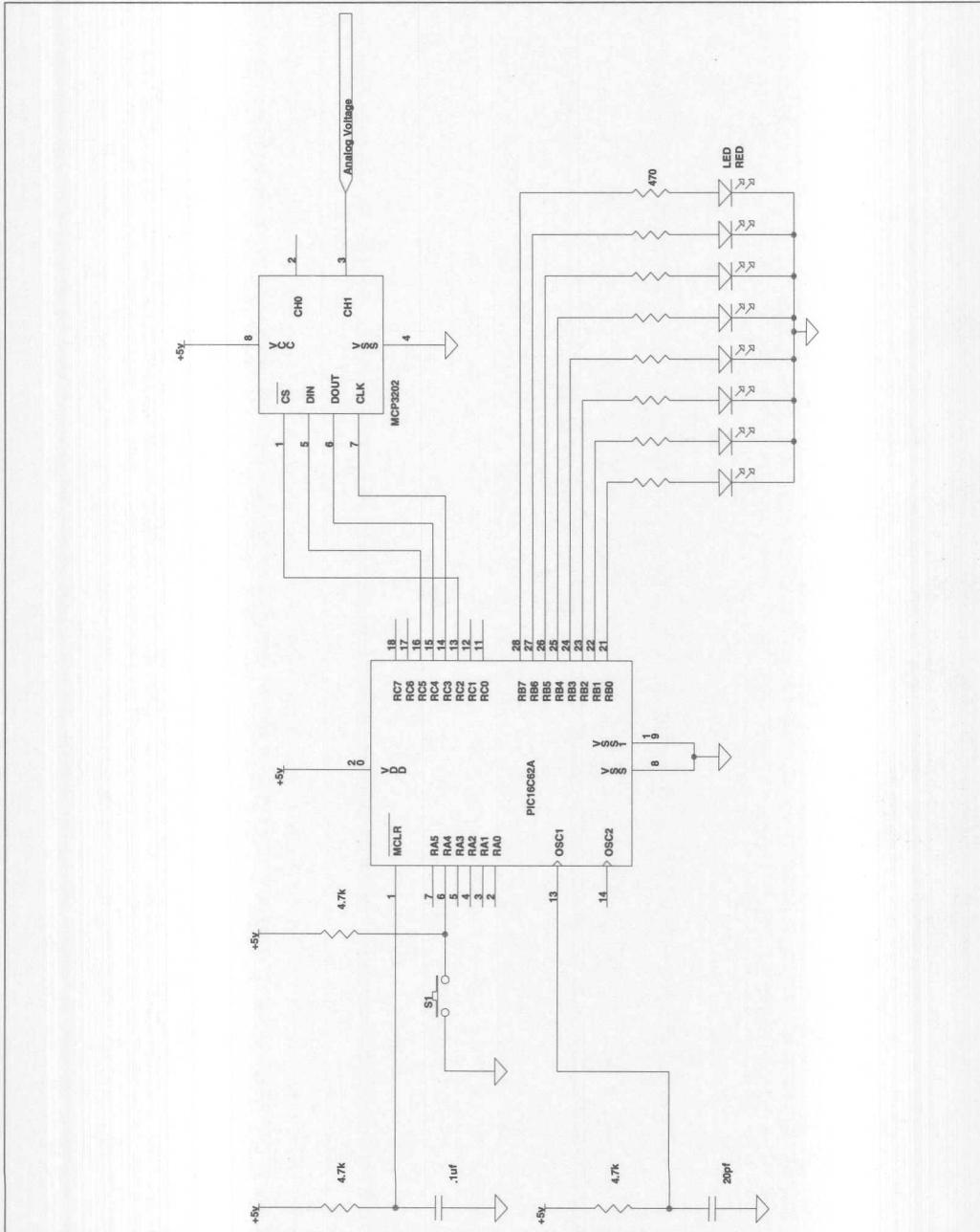
void main(void)
{
    union DualAccess
    {
        unsigned int By_16;      /* declare union to allow access to */
                                /* variable as 8 or 16-bit */      0x12 = CS1INT
                                /* allows 16-bit access */        0x13 = CS1INT
        struct Bytewise          /* struct provides for 8-bit access */
        {
            unsigned char Lo;    /* LSB of variable */          0x14 = CS1INT
            unsigned char Hi;    /* MSB of variable */          0x15 = CS1INT
        } By_8;
    } ADresult;
}

```

AN703

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX I: EQUIVALENT SCHEMATIC



AN703

NOTES:



MICROCHIP

AN704

Interfacing Microchip's MCP3201 Analog/Digital (A/D) Converter to MC68HC11E9-Based Microcontroller

Author: Richard L. Fischer
Microchip Technology Inc.

INTRODUCTION

Many of the embedded control systems designed today require some flavor of Analog-to-Digital (A/D) Converter. Embedded system applications such as Data Acquisition, Sensor Monitoring, and Instrumentation and Control all have varying A/D Converter requirements.

For the most part, these A/D Converter requirements are a combination of performance, cost, package size basis and availability. In some applications a microcontroller with an on-chip A/D Converter may meet the design requirements. Typically, these on-chip A/D Converter modules fit well into embedded applications which require a 10-35ksps A/D Converter. In other applications, a stand-alone A/D Converter is required for various performance reasons.

For those applications which require higher performance or remote sense capability, the Microchip MCP3201 12-bit A/D Converter fits very nicely.

The Microchip Technology Inc. MCP3201 employs a classic SAR architecture. The device uses an internal sample and hold capacitor to store the analog input while the conversion is taking place. Conversion rates of 100ksps are possible on the MCP3201. Minimum clock speed (10 kHz or 625sp, assuming 16 clocks) is a function of the capacitors used for the sample and hold.

The MCP3201 has a single pseudo-differential input. The (IN-) input is limited to $\pm 100\text{mV}$. This can be used to cancel small noise signals present on both the (IN+) and (IN-) inputs. This provides a means of rejecting noise when the (IN-) input is used to sense a remote signal ground. The (IN+) input can range from the (IN-) input to V_{REF} .

The reference voltage for the MCP3201 is applied to V_{REF} pin. V_{REF} determines the analog input voltage range and the LSB size, i.e.:

$$\text{LSB size} = \frac{V_{REF}}{2^{12}}$$

As the reference input is reduced, the LSB size is reduced accordingly.

Communication with the MCP3201 is accomplished using a standard SPI™ compatible serial interface. This interface allows direct connection to the serial ports of microcontrollers and digital signal processors.

The MCP3201 is suitable for use with a wide variety of microcontrollers from Microchip and others. This application note describes how to interface the MCP3201 with a Motorola MC68HC11 microcontroller. Application Note, AN702 covers microcontrollers based on the Intel 8051 architecture.

Figure 1 shows the hardware schematic for this interface. Appendix A contains a listing of the source code.

CIRCUIT DESCRIPTION

The serial interface of the Microchip MCP3201 A/D Converter has three wires, a serial clock input (DCLK), the serial data output (D_{out}) and the chip select input signal ($\overline{CS}/SHDN$). For this simple circuit interface, the Motorola MC68HC11E9 SPI port is used. PORTD:<4> is configured for the serial clock and PORTD:<2> is the data input to the microcontroller. The SPI clock rate for this application is set at 1 MHz.

The MC68HC11 is configured in the master mode with its CPOL and CPHA bits set to logic one (default setting on power-up).

A conversion is initiated with the high to low transition of $\overline{CS}/SHDN$ (active low). The chip select is generated by PORTD:<5> of the microcontroller. The device will sample the analog input from the rising edge of the first clock after \overline{CS} goes low for 1.5 clock cycles. On the falling edge of the second clock, the device will output a low null bit. With the next 12 clocks, the MCP3201 will output the result of the conversion with the MSB first (See Figure 2 and Figure 3). Data is always output from the device on the falling edge of the clock. If the device continues to receive clocks while $\overline{CS}/SHDN$ is low, the device will output the conversion LSB first. If more clocks are provided to the device while $\overline{CS}/SHDN$ is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

2

Application Notes

SPI is a trademark of Motorola Inc.

Microcontroller Application Note AN704

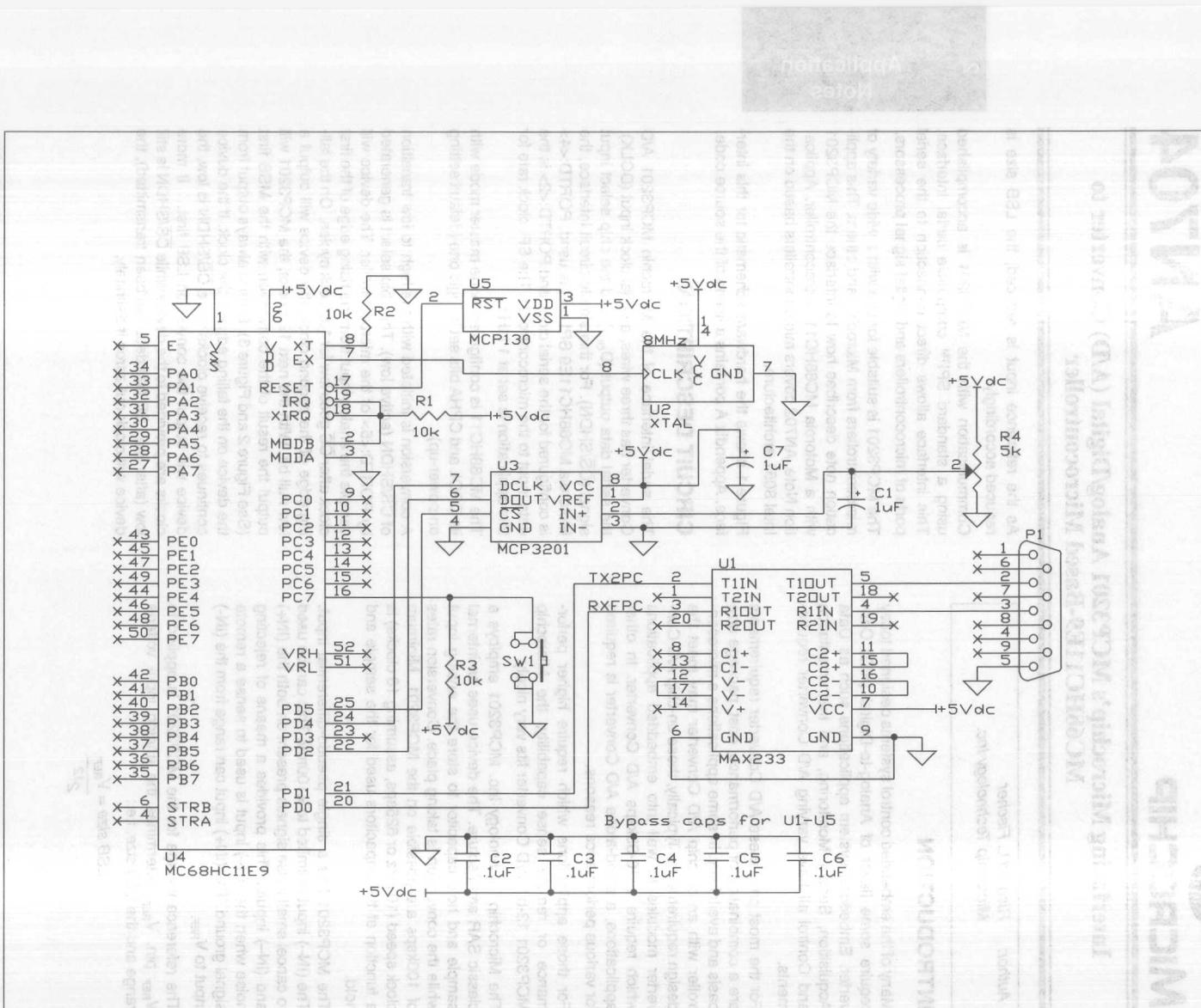


FIGURE 1: Hardware Schematic

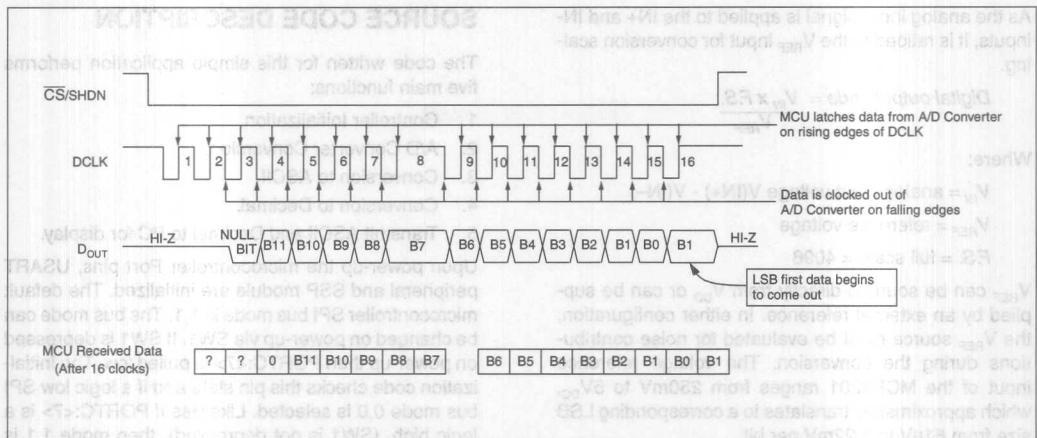


FIGURE 2: SPI Communication using 8-bit segments (Mode 1,1: SCLK idles high).

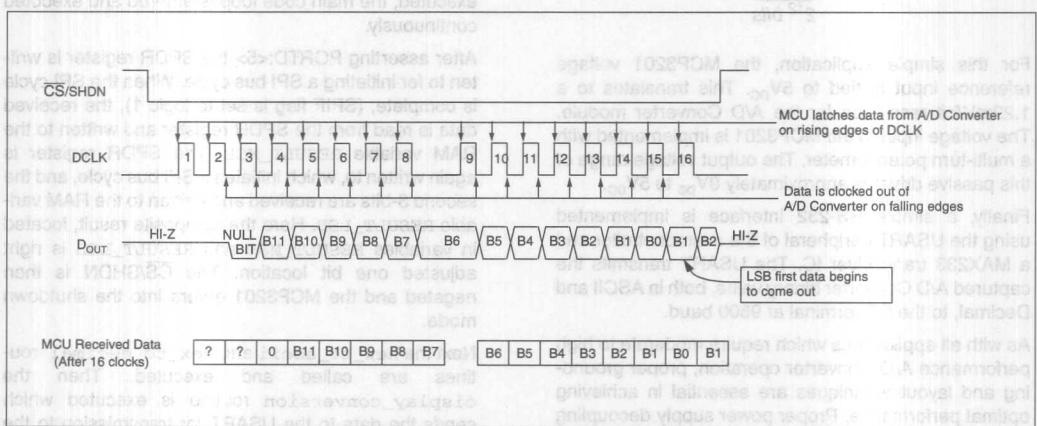


FIGURE 3: SPI Communication using 8-bit segments (Mode 0,0: SCLK idles low).

REFERENCES

- Williams, Jim, "Analog Choice: Delta-Sigma, Quantitative,"
- Heijnen, Peter, "Getting Started With 12-Bit TxD Counterless Applications," AN989, Microchip Technical Note
- M09850, 12-Bit A/D Converter with SPI Serial Interface DS21580B
- See, Microchip Technical Journal, "Introduction to

As the analog input signal is applied to the IN+ and IN- inputs, it is ratioed to the V_{REF} input for conversion scaling.

$$\text{Digital output code} = \frac{V_{IN} \times F.S.}{V_{REF}}$$

Where:

$$V_{IN} = \text{analog input voltage } V(\text{IN+}) - V(\text{IN-})$$

$$V_{REF} = \text{reference voltage}$$

$$F.S. = \text{full scale} = 4096$$

V_{REF} can be sourced directly from V_{DD} or can be supplied by an external reference. In either configuration, the V_{REF} source must be evaluated for noise contributions during the conversion. The voltage reference input of the MCP3201 ranges from 250mV to 5V_{DC}, which approximately translates to a corresponding LSB size from 61μV to 1.22mV per bit.

$$1.22\text{mV} = \frac{5V_{DC}}{2^{12} \text{ bits}}$$

For this simple application, the MCP3201 voltage reference input is tied to 5V_{DC}. This translates to a 1.22mV/bit resolution for the A/D Converter module. The voltage input to the MCP3201 is implemented with a multi-turn potentiometer. The output voltage range of this passive driver is approximately 0V_{DC} to 5V_{DC}.

Finally, a simple RS-232 interface is implemented using the USART peripheral of the microcontroller and a MAX233 transceiver IC. The USART transmits the captured A/D Converter binary value, both in ASCII and Decimal, to the PC terminal at 9600 baud.

As with all applications which require moderate to high performance A/D Converter operation, proper grounding and layout techniques are essential in achieving optimal performance. Proper power supply decoupling and input signal and V_{REF} parameters must be considered for noise contributions.

SOURCE CODE DESCRIPTION

The code written for this simple application performs five main functions:

1. Controller Initialization.
2. A/D Converter Conversion.
3. Conversion to ASCII.
4. Conversion to Decimal.
5. Transmit ASCII and Decimal to PC for display.

Upon power-up the microcontroller Port pins, USART peripheral and SSP module are initialized. The default microcontroller SPI bus mode is 1,1. The bus mode can be changed on power-up via SW1. If SW1 is depressed on power-up then PORTC:<7> is pulled low. The initialization code checks this pin state and if a logic low SPI bus mode 0,0 is selected. Likewise if PORTC:<7> is a logic high, (SW1 is not depressed), then mode 1,1 is the operational bus state. Once the initialization code is executed, the main code loop is entered and executed continuously.

After asserting PORTD:<5> the SPDR register is written to for initiating a SPI bus cycle. When the SPI cycle is complete, (SPIF flag is set to logic 1), the received data is read from the SPDR register and written to the RAM variable RESULT_MSB. The SPDR register is again written to, which initiates a SPI bus cycle, and the second 8-bits are received and written to the RAM variable RESULT_LSB. Here the composite result, located in variables RESULT_MSB and RESULT_LSB is right adjusted one bit location. The CS/SHDN is then negated and the MCP3201 enters into the shutdown mode.

Next the hex_to_ascii and hex_to_decimal routines are called and executed. Then the display_conversion routine is executed which sends the data to the USART for transmission to the PC for display.

REFERENCES

Williams, Jim, "Analog Circuit Design," Butterworth-Heinemann

Baker, Bonnie, "Layout Tips for 12-bit A/D Converter Applications," AN688, Microchip Technology Inc.

MCP3201 12-bit A/D Converter with SPI Serial Interface, Microchip Technology, Document DS21290B, 1999.

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX A: SOURCE CODE

MCP3201.SRC Assembled with IASM 07/01/1999 11:30 PAGE 1
Interfacing Microchip MCP3201 ADC to Motorola MC68HC11E9-Based Microcontroller

```

1
2
3 ****
4 *
5 *     Filename:      MCP3201.src
6 *     Date:          07/01/99
7 *     File Version:  1.00
8 *     Author:         Richard L. Fischer
9 *                  Microchip Technology Inc.
10 *
11 *
12 ****
13 ****
14
15
16 * This code demonstrates how the Microchip MCP3201 Analog-to-Digital
17 * Converter (ADC) is interfaced to the Synchronous Serial Peripheral
18 * (SSP) of the MC68HC11E9 microcontroller. The interface uses two
19 * Serial Peripheral Interface (SPI) lines (SCK, MISO) on the
20 * 68HC11E9 Microcontroller for the clock (SCK) and data in (MISO).
21 * A chip select (CS) to the MCP3201 is generated with a general
22 * purpose port line (PD5). The MC68HC11E9 is placed into the master
23 * mode which allows use of the port line PD5 for the CS control
24 * signal. The simple application uses Mode 00 or Mode 11 to the
25 * define bus clock polarity and phase.
26 *
27 * SPI bus mode 1,1 is the default mode of operation upon power-up.
28 * If SPI bus Mode 0,0 is desired, cycle off power, depress and
29 * hold SW1, cycle on power then release SW1. SPI bus Mode 0,0 is
30 * now the operational mode.
31 *
32 * For this application, the SPI data rate is set to one eighth of
33 * the microcontroller clock frequency. The MC68HC11E9 device clock
34 * frequency used for this application is 8MHz. This translates to
35 * an ADC throughput of 62.5kHz. In order to obtain the maximum
36 * throughput (100kHz) from the MCP3201 ADC the M68HC11E9 must be
37 * clocked at 12.8Mhz.
38 *
39 *
40 ****
41
42
43 ****
44 * MICROCONTROLLER RELATED EQUATES
45 ****
46
47 REGBASE EQU $1000 ; Register Base Address
48 PACTL  EQU $26 ; PortA bit7 control
49 PORTA  EQU $00 ; PortA Address
50 PORTB  EQU $04 ; PortB Address
51 DDRC   EQU $07 ; PortD Data Direction Register
52 PORTC  EQU $03 ; PortC Address
53 DDRD   EQU $09 ; PortD Data Direction Register
54 PORTD  EQU $08 ; PortD Address
55
56 SPCR   EQU $28 ; SPI Control Register
57 SPSR   EQU $29 ; SPI Status Register

```

```

0000      58  SPDR    EQU $2A      ; SPI Data Register
0000      59
0000      60  BAUD    EQU $2B      ; Baud Rate Register
0000      61  SCCR2   EQU $2D      ; SCI Control Register 2
0000      62  SCSR    EQU $2E      ; SCI Status Register
0000      63  SCDR    EQU $2F      ; SCI Data Register
0000      64
0000      65
0000      66  MASKCS  EQU $20      ; Mask for Chip Select Bit
0000      67
0000      68
0000      69  ****INTERNAL RAM USAGE****
0000      70  *           INTERNAL RAM USAGE
0000      71  ****INTERNAL RAM USAGE****
0000      72
0000      73  ORG $0000      ; Internal RAM
0000      74
0000      75  RESULT_MSB RMB 1      ; Converted MSB
0001      76  RESULT_LSB RMB 1      ; Converted LSB
0002      77
0002      78  THOUS   RMB 1      ; Variable for ASCII thousands
0003      79  HUNDS   RMB 1      ; Variable for ASCII hundreds
0004      80  TENS    RMB 1      ; Variable for ASCII tens
0005      81  ONES    RMB 1      ; Variable for ASCII ones
0006      82
0006      83  DISPLAY_BUFF RMB 1F  ; Buffer string for display
0006      84
0006      85
0006      86
0006      87  ****INTERNAL MEMORY EQUATES****
0006      88  *           INTERNAL MEMORY EQUATES
0006      89  ****INTERNAL MEMORY EQUATES****
0025      90
0025      91  EEPROMBEG EQU $B600 ; EEPROM begin
0025      92  EEPROMEND EQU $B7FF ; EEPROM end
0025      93
0025      94  EPROMBEG EQU $D000 ; EPROM begin
0025      95  EPROMEND EQU $FFFF ; EPROM end
0025      96
0025      97  RAMBEG   EQU $0000 ; RAM begin
0025      98  RAMEND   EQU $01FF ; RAM end
0025      99
0025     100
0025     101
0025     102  ****INTERNAL MEMORY EQUATES****
0000      103
0000      104  ORG EPRMBEG      ; Beginning of code
0000      105
D000 [03] 8E01FF 106  start    LDS #RAMEND      ; Initialize Stack Pointer
D003 [06] 7F0000 107  CLR RESULT_MSB      ; Clear ram variables
D006 [06] 7F0001 108  CLR RESULT_LSB      ; Clear ram variables
109
D009 [03] CE0006 110  LDX #DISPLAY_BUFF      ; Initialize X pointer to RAM
D00C [04] 18CED12D 111  LDY #DISPLAY_IMAGE      ; Initialize Y pointer to ROM
D010 [05] 18A600 112  copy     LDAA 0,Y      ; Read from ROM
D013 [04] A700 113  copy     STAA 0,X      ; Save into RAM
D015 [03] 08 114  incx    INX      ; Point to next RAM location
D016 [04] 1808 115  incy    INY      ; Point to next ROM location
D018 [04] 8C0026 116  cp      CPX #DISPLAY_BUFF+20 ; Copy display string complete?
D01B [03] 26F3 117  bne     BNE copy      ; No, so copy some more
118
119
D01D [03] CE1000 120  ldx     LDX #REGBASE      ; Initialize X Index pointer
121
D020 [07] 1D00FF 122  bclr    BCLR PORTA,X,$FF ; Set PortA outputs low
D023 [07] 1C2680 123  bset    BSET PACTL,X,$80 ; Set PortA bit7 to output

```

```

D026 [07] 1D04FF    124      BCLR  PORTB,X,$FF    ; Set PortB outputs low
D029 [07] 1C077F    125      BSET  DDRC,X,$7F    ; Set PortC for outputs
                                ; except for pin7
D02C [07] 1D03FF    126      BCLR  PORTC,X,$FF    ; Set PortC outputs low
D02F [07] 1C0937    127      BSET  DDRD,X,$37    ; Make all PortD pins outputs
                                ; except PD3
D032 [02] 8620        128      LDAA  #$20    ; Set CS high and all other
D034 [04] A708        129      STAA  PORTD,X
D036 [07] 1C2D08    130      LDAA  #$20    ; Set CS high and all other
D039 [02] 8630        131      STAA  PORTD,X
D03B [04] A72B        132      BSET  SCCR2,X,$08  ; Enable SCI TX
D03D [07] 1E038006   133      LDAA  #$30    ; Initialize SCI for 9600
D041 [02] 8650        134      STAA  BAUD,X    ; baud at 8Mhz
D043 [04] A728        135      BRA   start_conversion ; Go start conversion
D045 [03] 2004        136      BRA   start_conversion
D047 [02] 865C        137      BRSET PORTC,X,$80,m11 ; Branch to Mode 11 if pin high
D049 [04] A728        138      LDAA  #$50    ; Else, Mode 00
D04A [04] A728        139      STAA  SPCR,X    ; Initialize SPI control reg
D045 [03] 2004        140      BRA   start_conversion
D047 [02] 865C        141      BRA   start_conversion
D049 [04] A728        142      LDAA  #$5C    ; Mode 11
D04A [04] A728        143      STAA  SPCR,X    ; Initialize SPI control reg
D044 [04] A728        144      BRA   start_conversion
D045 [04] A728        145      LDAA  #00    ; X=0
D046 [04] A728        146      STAA  #00    ; X=0
D047 [04] A728        147      ***** The routines to follow perform 3 repetitive functions:
D048 [04] A728        148      *
D049 [04] A728        149      * 1. Initiate an ADC conversion sequence
D050 [07] 1D0820      150      *
D051 [06] 6F2A        151      * 2. Convert acquired data for display
D052 [07] 1F2980FC    152      * 3. Transmit converted data to PC
D053 [04] A62A        153      *
D054 [04] A62A        154      *
D055 [03] 9700        155      *****
D056 [04] A62A        156      LDAA  #00    ; del_ls
D057 [04] A62A        157      STAA  RESULT    ; start_conversion
D058 [06] 6F2A        158      LDAA  #00    ; Y=0
D059 [07] 1D0820      159      BCLR  PORTD,X,MASKCS ; Assert CS to ADC
D060 [06] 6F2A        160      CLR   SPDR,X    ; Initiate SPI bus cycle
D061 [06] 6F2A        161      LDAA  SPDR,X    ; Read data and clear flag (SPIF)
D062 [07] 1F2980FC    162      BRCLR SPSPR,X,$80,rd1     ; Wait until cycle completes
D063 [04] A62A        163      LDAA  SPDR,X    ; Read data and clear flag (SPIF)
D064 [04] A62A        164      STAA  RESULT_MSB ; Save off upper bits
D065 [06] 760000      165      LDAA  #00    ; del_ls
D066 [06] 760001      166      CLR   SPDR,X    ; Initiate SPI bus cycle
D067 [06] 1F2980FC    167      BRCLR SPSPR,X,$80,rd2     ; Wait until cycle completes
D068 [07] 1C0820      168      BSET  PORTD,X,MASKCS ; Negate CS, place ADC in shutdown
D069 [04] A62A        169      LDAA  SPDR,X    ; Read data and clear flag (SPIF)
D070 [04] A62A        170      STAA  RESULT_LSB ; Save off lower bits
D071 [06] 1500FO      171      ROR   RESULT_MSB ; Move bit 0 into carry
D072 [06] 1500FO      172      ROR   RESULT_LSB ; Rotate MSB bit0 into LSB bit7
D073 [06] 1500FO      173      BCLR  RESULT_MSB,$F0 ; Mask out upper bits of MSB
D074 [06] BDD093      174      LDAA  #00    ; del_ls
D075 [06] BDD0D3      175      JSR   hex_to_ascii ; Convert to ASCII for display
D076 [06] BDD0D3      176      JSR   hex_to_decimal ; Convert to decimal for display
D077 [06] BDD0D3      177      JSR   hex_to_ascii ; Convert to ASCII for display
D078 [06] BDD0D3      178      JSR   hex_to_decimal ; Convert to decimal for display
D079 [06] BDD0D3      179      JSR   hex_to_ascii ; Convert to ASCII for display
D080 [06] BDD0D3      180      JSR   hex_to_decimal ; Convert to decimal for display
D081 [06] BDD0D3      181      ***** ROUTINE FOR DISPLAYING CONVERTED DATA *****
D082 [06] BDD0D3      182      JSR   display_conversion
D083 [06] BDD0D3      183      JSR   hex_to_ascii ; Initialize Y pointer
D084 [06] BDD0D3      184      JSR   hex_to_decimal ; Initialize Y pointer
D085 [06] BDD0D3      185      JSR   hex_to_ascii ; Initialize Y pointer
D086 [06] BDD0D3      186      LDY   #DISPLAY_BUFF ; Initialize Y pointer
D087 [05] 18A600      187      LDAA  0,Y    ; Retrieve MSB upper nibble
D088 [04] A72F        188      BCLR  RESULT_MSB ; Initiate SCI Transmit
D089 [04] A72F        189      STAA  SCDR,X    ; Initiate SCI Transmit

```

AN704

```

D07D [07] 1F2E40FC 190 wtx 000 BRCLR SCSR,X,$40,wtx ; Wait until cycle completes [0] 8000
D081 [04] 1808 000 191 00000000 INY #DISPLAY_BUFF+20 ; Increment Y pointer [0] 8000
D083 [05] 188C0026 192 00000000 CPY #DISPLAY_BUFF+20 ; All characters sent? [0] 8000
D087 [03] 26EF 000 193 00000000 BNE tx_loop ; No, so send more characters [0] 8000
D089 [04] 18CE0000 194 00000000 195 00000000
D08D [04] 1809 000 196 LDY #$0000 ; Approximately 485mS delay [0] 8000
D08F [03] 26FC 197 decy 000 DEY ; Decrement counter [0] 8000
D091 [03] 20B8 000 198 BNE decy ; Stay in loop until Y=0 [0] 8000
D093 [04] 3C 199 00000000
D094 [03] CE0000 200 00000000 BRA start_conversion ; Stay in main loop [0] 8000
D097 [04] 18CE000E 201 00000000
D099 [04] A600 202 00000000
D0A0 [02] 84F0 203 00000000 ***** ROUTINE FOR CONVERTING TO ASCII *****
D0A1 [02] 44 204 00000000
D0A2 [02] 44 205 00000000
D0A3 [02] 44 206 hex_to_ascii 00000000
D0A4 [02] 8109 207 00000000
D0A6 [03] 2F07 208 00000000 PSHX ; Save off X Index register [0] 8000
D0A7 [02] 8B37 209 00000000 LDX #RESULT_MSB ; Initialize X [0] 8000
D0A9 [04] 18A700 210 00000000 LDY #DISPLAY_BUFF+8 ; Initialize Y [0] 8000
D0A9B [04] A600 211 00000000 LDAA 0,X ; Get Result_MSB data [0] 8000
D0A9D [03] 36 212 00000000 PSHA ; Save ACCA [0] 8000
D0A9E [02] 84F0 213 00000000 ANDA #$F0 ; Mask out lower nibble [0] 8000
D0A9F [02] 44 214 00000000 LSRA ; Shift upper nibble [0] 8000
D0A9G [02] 44 215 00000000 LSRA ; into lower nibble [0] 8000
D0A9H [02] 44 216 00000000 ; [0] 8000
D0A9I [02] 44 217 00000000 LSRA ; [0] 8000
D0A9J [02] 8109 218 00000000 CMPA #$09 ; Is value a number ? [0] 8000
D0A9K [02] 2F07 219 00000000 BLE numb_1 ; Yes, make it 0-9 ($30 - $39) [0] 8000
D0A9L [02] 8B37 220 00000000 ADDA #$37 ; No, make it a letter (A - F) [0] 8000
D0A9M [05] 18A700 221 00000000 STAA 0,Y ; Save ASCII letter [0] 8000
D0A9N [03] 2005 222 00000000 BRA do_lsb ; Convert lower nibble [0] 8000
D0A9O [02] 8B30 223 00000000 ADDA #$30 ; [0] 8000
D0A9P [05] 18A700 224 00000000 STAA 0,Y ; Save ASCII number [0] 8000
D0A9Q [04] 1808 225 00000000
D0A9R [04] 32 226 do_lsb INY ; Increment Y [0] 8000
D0A9S [04] 840F 227 00000000 PULA ; Restore ACCA [0] 8000
D0A9T [02] 840F 228 00000000 ANDA #$0F ; Mask out upper nibble [0] 8000
D0A9U [02] 8109 229 00000000 CMPA #$09 ; Is value a number ? [0] 8000
D0A9V [03] 2F07 230 00000000 BLE numb_2 ; Yes, make it 0-9 ($30 - $39) [0] 8000
D0A9W [02] 8B37 231 00000000 ADDA #$37 ; No, make it a letter (A - F) [0] 8000
D0A9X [05] 18A700 232 00000000 STAA 0,Y ; Save ASCII letter [0] 8000
D0A9Y [03] 2005 233 00000000 BRA next1 ; [0] 8000
D0A9Z [04] 1808 234 00000000
D0A9[05] 18A700 235 00000000 ADDA #$30 ; Save ASCII number [0] 8000
D0A9[06] 18A700 236 00000000 STAA 0,Y ; Save ASCII number [0] 8000
D0A9[07] 18A700 237 00000000 ; [0] 8000
D0A9[08] 08 238 next1 INX ; Increment X [0] 8000
D0A9[09] 1808 239 00000000 INY ; Increment Y [0] 8000
D0A9[0A] 8C0002 240 00000000 CPX #RESULT_MSB+2 ; Converted all bytes? [0] 8000
D0A9[0B] 26CA 241 00000000 BNE next1 ; No, so do some more [0] 8000
D0A9[0C] 38 242 00000000 PULX ; Else yes so restore X [0] 8000
D0A9[0D] 39 243 00000000 RTS ; Return from subroutine [0] 8000
D0A9[0E] 39 244 00000000
D0A9[0F] 245 00000000
D0A9[10] 246 00000000
D0A9[11] 247 ***** ROUTINE FOR CONVERTING TO DECIMAL *****
D0A9[12] 7F0002 248 00000000
D0A9[13] 7F0003 249 hex_to_decimal 00000000
D0A9[14] 7F0004 250 00000000
D0A9[15] 7F0005 251 00000000 LDD RESULT_MSB ; Initialize ACCA and ACCB [0] 8000
D0A9[16] 7F0006 252 00000000 CLR THOUS ; Zero out THOUSANDS temp location [0] 8000
D0A9[17] 7F0007 253 00000000 CLR HUND ; Zero out HUNDREDS temp location [0] 8000
D0A9[18] 7F0008 254 00000000 CLR TENS ; Zero out TENS temp location [0] 8000
D0A9[19] 7F0009 255 00000000 CLR ONES ; Zero out ONES temp location [0] 8000

```

```

256      ; ACCD >= 1000 or more?
257  ck_thous CPD #$03E7 ; No, so go check hundreds
258      BLO ck_hunds ; Subtract 1000 from ACCD
259      SUBD #$03E8 ; Increment THOUS
260      INC THOUS ; Go check for more
261      BRA ck_thous
262
263      ck_hunds CPD #$0063 ; ACCD >= 100 or more?
264      BLO ck_tens ; No, so go check tens
265      SUBD #$0064 ; Subtract 100 from ACCD
266      INC HUND ; Increment HUND
267      BRA ck_hunds ; Go check for more
268
269      ck_tens CMPB #$0A ; No, ACCB >= 10 or more?
270      BLO do_ones ; No, finish up with ONES
271      SUBB #$0A ; Subtract another 10 from B
272      INC TENS ; Bump "TENS"
273      BRA ck_tens ; Loop again
274
275      do_ones ADDB #$30 ; Convert ONES to ASCII
276      LDY #DISPLAY_BUFF+18 ; Initialize Y pointer into buffer
277      LDAA THOUS ; 
278      ADDA #$30 ; Convert THOUS to ASCII
279      STAA 0,Y ; Save it
280      INY ; Increment Y
281      LDAA HUND ; 
282      ADDA #$30 ; Convert HUND to ASCII
283      STAA 0,Y ; Save it
284      INY ; Increment Y
285      LDAA TENS ; 
286      ADDA #$30 ; Convert TENS to ASCII
287      STAA 0,Y ; Save it
288      INY ; Increment Y
289      STAB 0,Y ; Save conversion into string
290      RTS ; Return from subroutine
291
292
293
D12D    4865782D : Decimal-> '0D,0A
3E203078
20202020
203A2044
6563696D
616C2D3E
20202020
2000DA
294  DISPLAY_IMAGE DB 'Hex-> 0x
295
296
297 ****
298 *      INTERRUPT VECTORS
299 ****
300
FFD6    301      ORG $FFD6 ; VECTORS
302
FFD6    D000 303 DW start ; SCI Serial System - RIE, TIE, TCIE, ILIE
FFD8    D000 304 DW start ; SPI Serial Transfer Complete - SPIE
FFDA    D000 305 DW start ; Pulse Accumulator Input Edge - PAII
FFDC    D000 306 DW start ; Pulse Accumulator Overflow - PAOVI
FFDE    D000 307 DW start ; Timer Overflow - TOI
FFE0    D000 308 DW start ; Timer Input Capture 4/Output Compare 5 - I4/O5I
FFE2    D000 309 DW start ; Timer Output Compare 4 - OC4I
FFE4    D000 310 DW start ; Timer Output Compare 3 - OC3I
FFE6    D000 311 DW start ; Timer Output Compare 2 - OC2I
FFE8    D000 312 DW start ; Timer Output Compare 1 - OC1I
FFEA    D000 313 DW start ; Timer Input Capture 3 - IC3I
FFEC    D000 314 DW start ; Timer Input Capture 2 - IC2I
FFEE    D000 315 DW start ; Timer Input Capture 1 - IC1I

```




Controller Area Network (CAN) Basics

Author: Keith Pazul
Microchip Technology Inc.

INTRODUCTION

Controller Area Network (CAN) was initially created by German automotive system supplier Robert Bosch in the mid-1980s for automotive applications as a method for enabling robust serial communication. The goal was to make automobiles more reliable, safe and fuel-efficient while decreasing wiring harness weight and complexity. Since its inception, the CAN protocol has gained widespread popularity in industrial automation and automotive/truck applications. Other markets where networked solutions can bring attractive benefits like medical equipment, test equipment and mobile machines are also starting to utilize the benefits of CAN. The goal of this application note is to explain some of the basics of CAN and show the benefits of choosing CAN for embedded systems networked applications.

CAN OVERVIEW

Most network applications follow a layered approach to system implementation. This systematic approach enables interoperability between products from different manufacturers. A standard was created by the International Standards Organization (ISO) as a template to follow for this layered approach. It is called the ISO Open Systems Interconnection (OSI) Network Layering Reference Model and is shown in Figure 1 for reference.

The CAN protocol itself implements most of the lower two layers of this reference model. The communication medium portion of the model was purposely left out of the Bosch CAN specification to enable system designers to adapt and optimize the communication protocol on multiple media for maximum flexibility (twisted pair, single wire, optically isolated, RF, IR, etc.). With this flexibility, however, comes the possibility of interoperability concerns.

To ease some of these concerns, the International Standards Organization and Society of Automotive Engineers (SAE) have defined some protocols based on CAN that include the Media Dependant Interface definition such that all of the lower two layers are specified.

ISO11898 is a standard for high-speed applications, ISO11519 is a standard for low-speed applications, and J1939 (from SAE) is targeted for truck and bus applications. All three of these protocols specify a 5V differential electrical bus as the physical interface.

The rest of the layers of the ISO/OSI protocol stack are left to be implemented by the system software developer. Higher Layer Protocols (HLPs) are generally used to implement the upper five layers of the OSI Reference Model.

HLPs are used to:

- 1) standardize startup procedures including bit rates used,
- 2) distribute addresses among participating nodes or types of messages,
- 3) determine the structure of the messages, and
- 4) provide system-level error handling routines.

This is by no means a full list of the functions HLPs perform, however it does describe some of their basic functionality.

CAN PROTOCOL BASICS

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

The CAN communication protocol is a CSMA/CD protocol. The CSMA stands for Carrier Sense Multiple Access. What this means is that every node on the network must monitor the bus for a period of no activity before trying to send a message on the bus (Carrier Sense). Also, once this period of no activity occurs, every node on the bus has an equal opportunity to transmit a message (Multiple Access). The CD stands for Collision Detection. If two nodes on the network start transmitting at the same time, the nodes will detect the 'collision' and take the appropriate action. In CAN protocol, a non-destructive bitwise arbitration method is utilized. This means that messages remain intact after arbitration is completed even if collisions are detected. All of this arbitration takes place without corruption or delay of the higher priority message.

There are a couple of things that are required to support non-destructive bitwise arbitration. First, logic states need to be defined as dominant or recessive. Second, the transmitting node must monitor the state of the bus to see if the logic state it is trying to send actually appears on the bus. CAN defines a logic bit 0 as a dominant bit and a logic bit 1 as a recessive bit.

A dominant bit state will always win arbitration over a recessive bit state, therefore the lower the value in the Message Identifier (the field used in the message arbitration process), the higher the priority of the message. As an example, suppose two nodes are trying to transmit a message at the same time. Each node will monitor the bus to make sure the bit that it is trying to send actually appears on the bus. The lower priority message will at some point try to send a recessive bit and the monitored state on the bus will be a dominant. At that point this node loses arbitration and immediately stops transmitting. The higher priority message will continue until completion and the node that lost arbitration will wait for the next period of no activity on the bus and try to transmit its message again.

Message-Based Communication

CAN protocol is a message-based protocol, not an address based protocol. This means that messages are not transmitted from one node to another node based on addresses. Embedded in the CAN message itself is the priority and the contents of the data being transmitted. All nodes in the system receive every message transmitted on the bus (and will acknowledge if the message was properly received). It is up to each node in the system to decide whether the message received should be immediately discarded or kept to be processed. A single message can be destined for one particular node to receive, or many nodes based on the way the network and system are designed.

For example, an automotive airbag sensor can be connected via CAN to a safety system router node only. This router node takes in other safety system information and routes it to all other nodes on the safety system network. Then all the other nodes on the safety system network can receive the latest airbag sensor information from the router at the same time, acknowledge if the message was received properly, and decide whether to utilize this information or discard it.

Another useful feature built into the CAN protocol is the ability for a node to request information from other nodes. This is called a Remote Transmit Request (RTR). This is different from the example in the previous paragraph because instead of waiting for information to be sent by a particular node, this node specifically requests data to be sent to it.

For example, a safety system in a car gets frequent updates from critical sensors like the airbags, but it may not receive frequent updates from other sensors like the oil pressure sensor or the low battery sensor to make sure they are functioning properly. Periodically, the safety system can request data from these other sensors and perform a thorough safety system check. The system designer can utilize this feature to minimize network traffic while still maintaining the integrity of the network.

One additional benefit of this message-based protocol is that additional nodes can be added to the system without the necessity to reprogram all other nodes to recognize this addition. This new node will start receiv-

ing messages from the network and, based on the message ID, decide whether to process or discard the received information.

CAN Message Frame Description

CAN protocol defines four different types of messages (or Frames). The first and most common type of frame is a Data Frame. This is used when a node transmits information to any or all other nodes in the system. Second is a Remote Frame, which is basically a Data Frame with the RTR bit set to signify it is a Remote Transmit Request (see Figure 2 and Figure 3 for details on Data Frames). The other two frame types are for handling errors. One is called an Error Frame and one is called an Overload Frame. Error Frames are generated by nodes that detect any one of the many protocol errors defined by CAN. Overload errors are generated by nodes that require more time to process messages already received.

Data Frames consist of fields that provide additional information about the message as defined by the CAN specification. Embedded in the Data Frames are Arbitration Fields, Control Fields, Data Fields, CRC Fields, a 2-bit Acknowledge Field and an End of Frame.

The Arbitration Field is used to prioritize messages on the bus. Since the CAN protocol defines a logical 0 as the dominant state, the lower the number in the arbitration field, the higher priority the message has on the bus. The arbitration field consists of 12-bits (11 identifier bits and one RTR bit) or 32-bits (29 identifier bits, 1-bit to define the message as an extended data frame, an SRR bit which is unused, and an RTR bit), depending on whether Standard Frames or Extended Frames are being utilized. The current version of the CAN specification, version 2.0B, defines 29-bit identifiers and calls them Extended Frames. Previous versions of the CAN specification defined 11-bit identifiers which are called Standard Frames.

As described in the preceding section, the Remote Transmit Request (RTR) is used by a node when it requires information to be sent to it from another node. To accomplish an RTR, a Remote Frame is sent with the identifier of the required Data Frame. The RTR bit in the Arbitration Field is utilized to differentiate between a Remote Frame and a Data Frame. If the RTR bit is recessive, then the message is a Remote Frame. If the RTR bit is dominant, the message is a Data Frame.

The Control Field consists of six bits. The MSB is the IDE bit (signifies Extended Frame) which should be dominant for Standard Data Frames. This bit determines if the message is a Standard or Extended Frame. In Extended Frames, this bit is RB1 and it is reserved. The next bit is RB0 and it is also reserved. The four LSbs are the Data Length Code (DLC) bits. The Data Length Code bits determine how many data bytes are included in the message. It should be noted that a Remote Frame has no data field, regardless of the value of the DLC bits.

The Data Field consists of the number of data bytes described in the Data Length Code of the Control Field.

The CRC Field consists of a 15-bit CRC field and a CRC delimiter, and is used by receiving nodes to determine if transmission errors have occurred.

The Acknowledge Field is utilized to indicate if the message was received correctly. Any node that has correctly received the message, regardless of whether the node processes or discards the data, puts a dominant bit on the bus in the ACK Slot bit time (see Figure 2 or Figure 3 for the location of the ACK Slot bit time).

The last two message types are Error Frames and Overload Frames. When a node detects one of the many types of errors defined by the CAN protocol, an Error Frame occurs. Overload Frames tell the network that the node sending the Overload Frame is not ready to receive additional messages at this time, or that intermission has been violated. These errors will be discussed in more detail in the next section.

Fast, Robust Communication

Because CAN was initially designed for use in automobiles, a protocol that efficiently handled errors was critical if it was to gain market acceptance. With the release of version 2.0B of the CAN specification, the maximum communication rate was increased 8x over the version 1.0 specification to 1Mbit/sec. At this rate, even the most time-critical parameters can be transmitted serially without latency concerns. In addition to this, the CAN protocol has a comprehensive list of errors it can detect that ensures the integrity of messages.

CAN nodes have the ability to determine fault conditions and transition to different modes based on the severity of problems being encountered. They also have the ability to detect short disturbances from permanent failures and modify their functionality accordingly. CAN nodes can transition from functioning like a normal node (being able to transmit and receive messages normally), to shutting down completely (bus-off) based on the severity of the errors detected. This feature is called Fault Confinement. No faulty CAN node or nodes will be able to monopolize all of the bandwidth on the network because faults will be confined to the faulty nodes and these faulty nodes will shut off before bringing the network down. This is very powerful because Fault Confinement guarantees bandwidth for critical system information.

As discussed previously, there are five error conditions that are defined in the CAN protocol and three error states that a node can be in, based upon the type and number of error conditions detected. The following section describes each one in more detail.

Errors Detected

CRC Error

A 15-bit Cyclic Redundancy Check (CRC) value is calculated by the transmitting node and this 15-bit value is transmitted in the CRC field. All nodes on the network receive this message, calculate a CRC and verify that the CRC values match. If the values do not match, a CRC error occurs and an Error Frame is generated. Since at least one node did not properly receive the message, it is then resent after a proper intermission time.

Acknowledge Error

In the Acknowledge Field of a message, the transmitting node checks if the Acknowledge Slot (which it has sent as a recessive bit) contains a dominant bit. This dominant bit would acknowledge that at least one node correctly received the message. If this bit is recessive, then no node received the message properly. An Acknowledge Error has occurred. An Error Frame is then generated and the original message will be repeated after a proper intermission time.

Form Error

If any node detects a dominant bit in one of the following four segments of the message: End of Frame, Interframe Space, Acknowledge Delimiter or CRC Delimiter, the CAN protocol defines this to be a form violation and a Form Error is generated. The original message is then resent after a proper intermission time. (see Figure 2 and/or Figure 3 for where these segments lie in a CAN message).

Bit Error

A Bit Error occurs if a transmitter sends a dominant bit and detects a recessive bit, or if it sends a recessive bit and detects a dominant bit when monitoring the actual bus level and comparing it to the bit that it has just sent. In the case where the transmitter sends a recessive bit and a dominant bit is detected during the Arbitration Field or Acknowledge Slot, no Bit Error is generated because normal arbitration or acknowledgment is occurring. If a Bit Error is detected, an Error Frame is generated and the original message is resent after a proper intermission time.

Stuff Error

CAN protocol uses a Non-Return-to-Zero (NRZ) transmission method. This means that the bit level is placed on the bus for the entire bit time. CAN is also asynchronous, and bit stuffing is used to allow receiving nodes to synchronize by recovering clock information from the data stream. Receiving nodes synchronize on recessive to dominant transitions. If there are more than five bits of the same polarity in a row, CAN will automatically stuff an opposite polarity bit in the data stream. The receiving node(s) will use

it for synchronization, but will ignore the stuff bit for data purposes. If, between the Start of Frame and the CRC Delimiter, six consecutive bits with the same polarity are detected, then the bit stuffing rule has been violated. A Stuff Error then occurs, an Error Frame is sent, and the message is repeated.

Error States

Detected errors are made public to all other nodes via Error Frames or Error Flags. The transmission of an erroneous message is aborted and the frame is repeated as soon as the message can again win arbitration on the network. Also, each node is in one of three error states, Error-Active, Error-Passive or Bus-Off.

Error-Active

An Error-Active node can actively take part in bus communication, including sending an active error flag, which consists of six consecutive dominant bits. The Error Flag actively violates the bit stuffing rule and causes all other nodes to send an Error Flag, called the Error Echo Flag, in response. An Active Error Flag, and the subsequent Error Echo Flag may cause as many as twelve consecutive dominant bits on the bus; six from the Active Error Flag, and zero up to six more from the Error Echo Flag depending upon when each node detects an error on the bus. A node is Error-Active when both the Transmit Error Counter (TEC) and the Receive Error Counter (REC) are below 128. Error-Active is the normal operational mode, allowing the node to transmit and receive without restrictions.

Error-Passive

A node becomes Error-Passive when either the Transmit Error Counter or Receive Error Counter exceeds 127. Error-Passive nodes are not permitted to transmit Active Error Flags on the bus, but instead, transmit Passive Error Flags which consist of six recessive bits. If the Error-Passive node is currently the only transmitter on the bus then the passive error flag will violate the bit stuffing rule and the receiving node(s) will respond with Error Flags of their own (either active or passive depending upon their own error state). If the Error-Passive node in question is not the only transmitter (i.e. during arbitration) or is a receiver, then the Passive Error Flag will have no effect on the bus due to the recessive nature of the error flag. When an Error-Passive node transmits a Passive Error Flag and detects a dominant bit, it must see the bus as being idle for eight additional bit times after an intermission before recognizing the bus as available. After this time, it will attempt to retransmit.

Bus-Off

A node goes into the Bus-Off state when the Transmit Error Counter is greater than 255 (receive errors can not cause a node to go Bus-Off). In this mode,

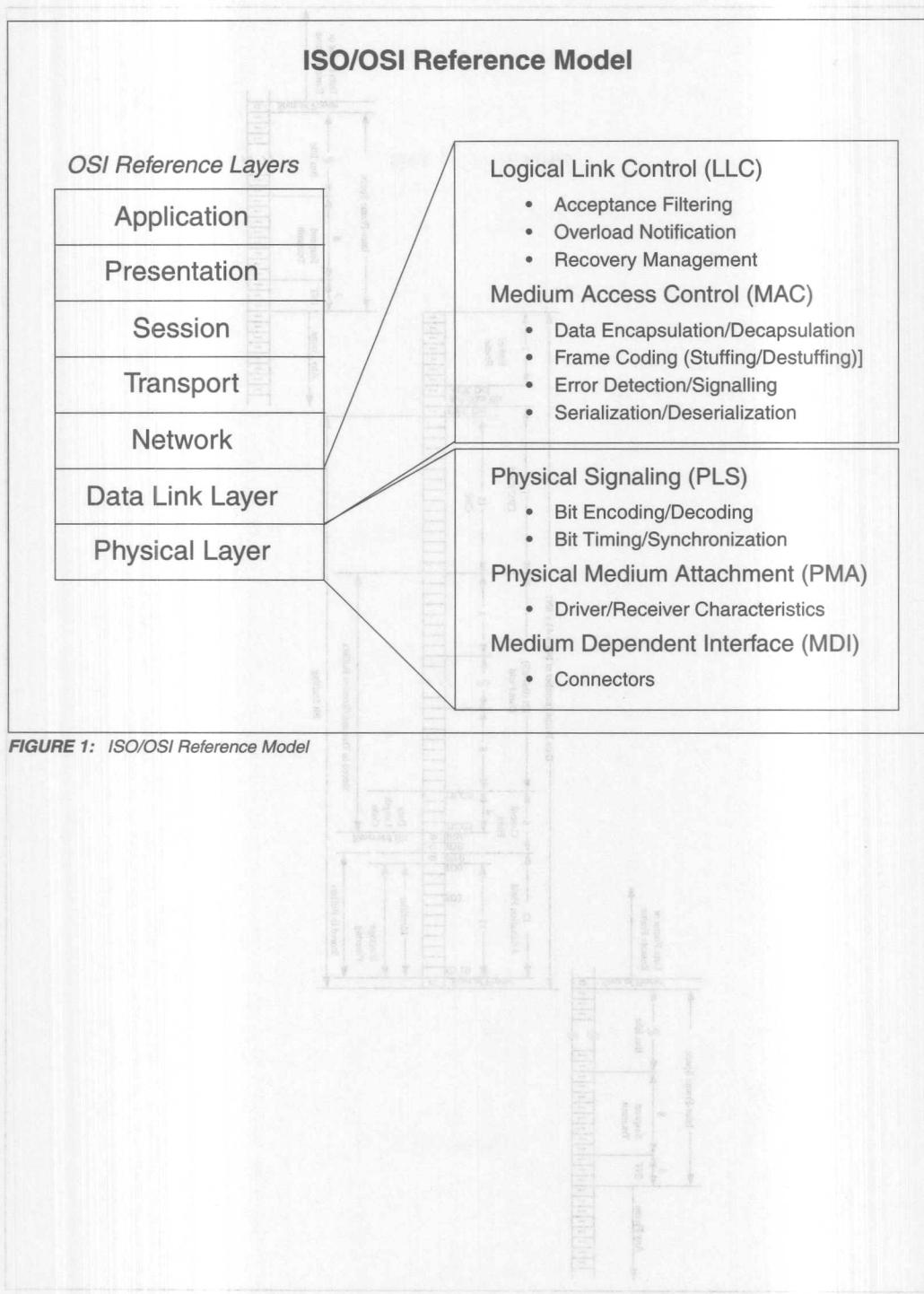
the node can not send or receive messages, acknowledge messages, or transmit Error Frames of any kind. This is how Fault Confinement is achieved. There is a bus recovery sequence that is defined by the CAN protocol that allows a node that is Bus-Off to recover, return to Error-Active, and begin transmitting again if the fault condition is removed.

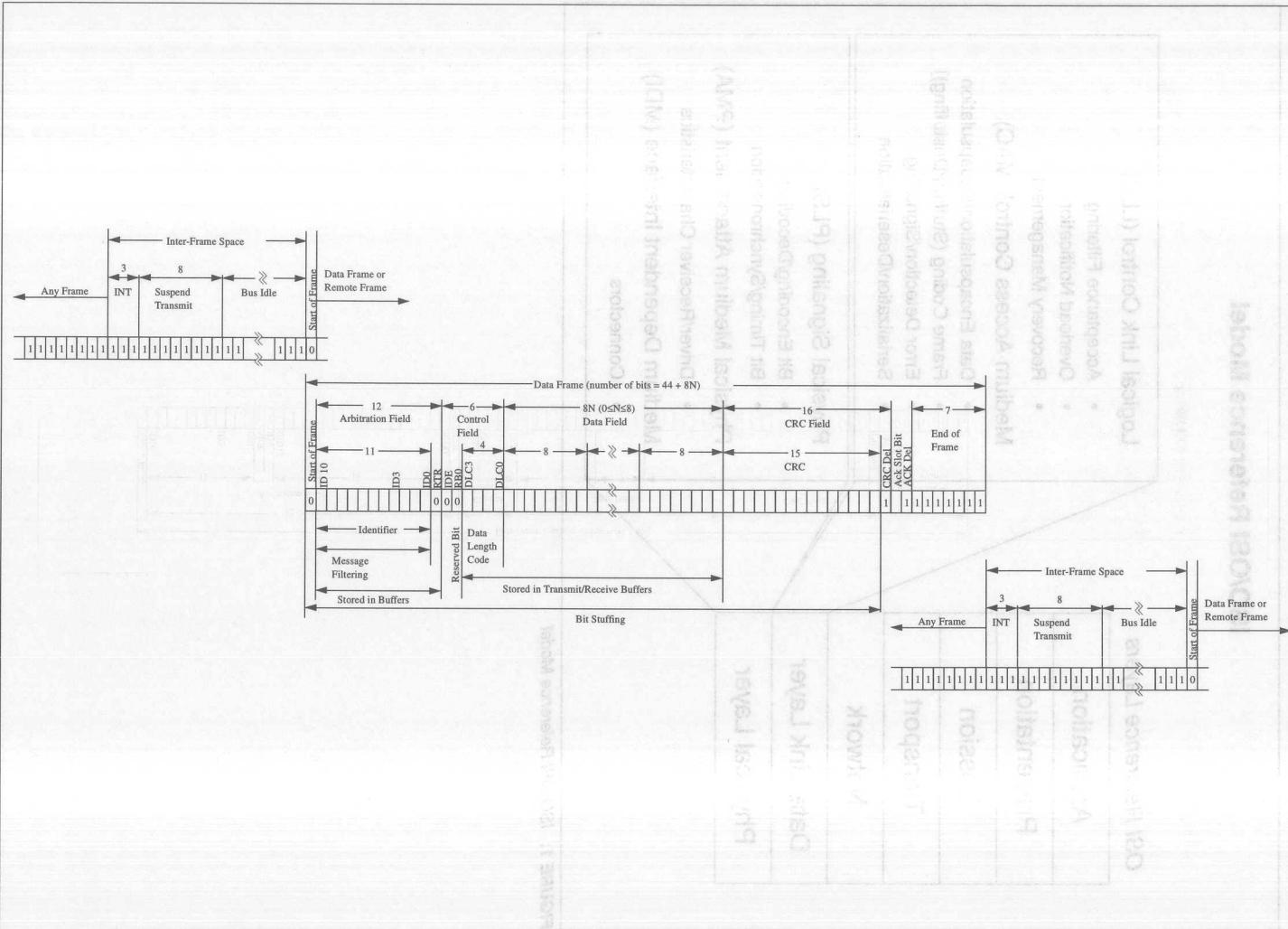
CONCLUSION

The CAN protocol was optimized for systems that need to transmit and receive relatively small amounts of information (as compared to Ethernet or USB, which are designed to move much larger blocks of data) reliably to any or all other nodes on the network. CSMA/CD allows every node to have an equal chance to gain access to the bus, and allows for smooth handling of collisions.

Since the protocol is message-based, not address based, all messages on the bus receive every message and acknowledge every message, regardless of whether it needs the data or not. This allows the bus to operate in node-to-node or multicast messaging formats without having to send different types of messages.

Fast, robust message transmission with fault confinement is also a big plus for CAN because faulty nodes will automatically drop off the bus not allowing any one node from bringing a network down. This effectively guarantees that bandwidth will always be available for critical messages to be transmitted. With all of these benefits built into the CAN protocol and its momentum in the automotive world, other markets will begin to see and implement CAN into their systems.





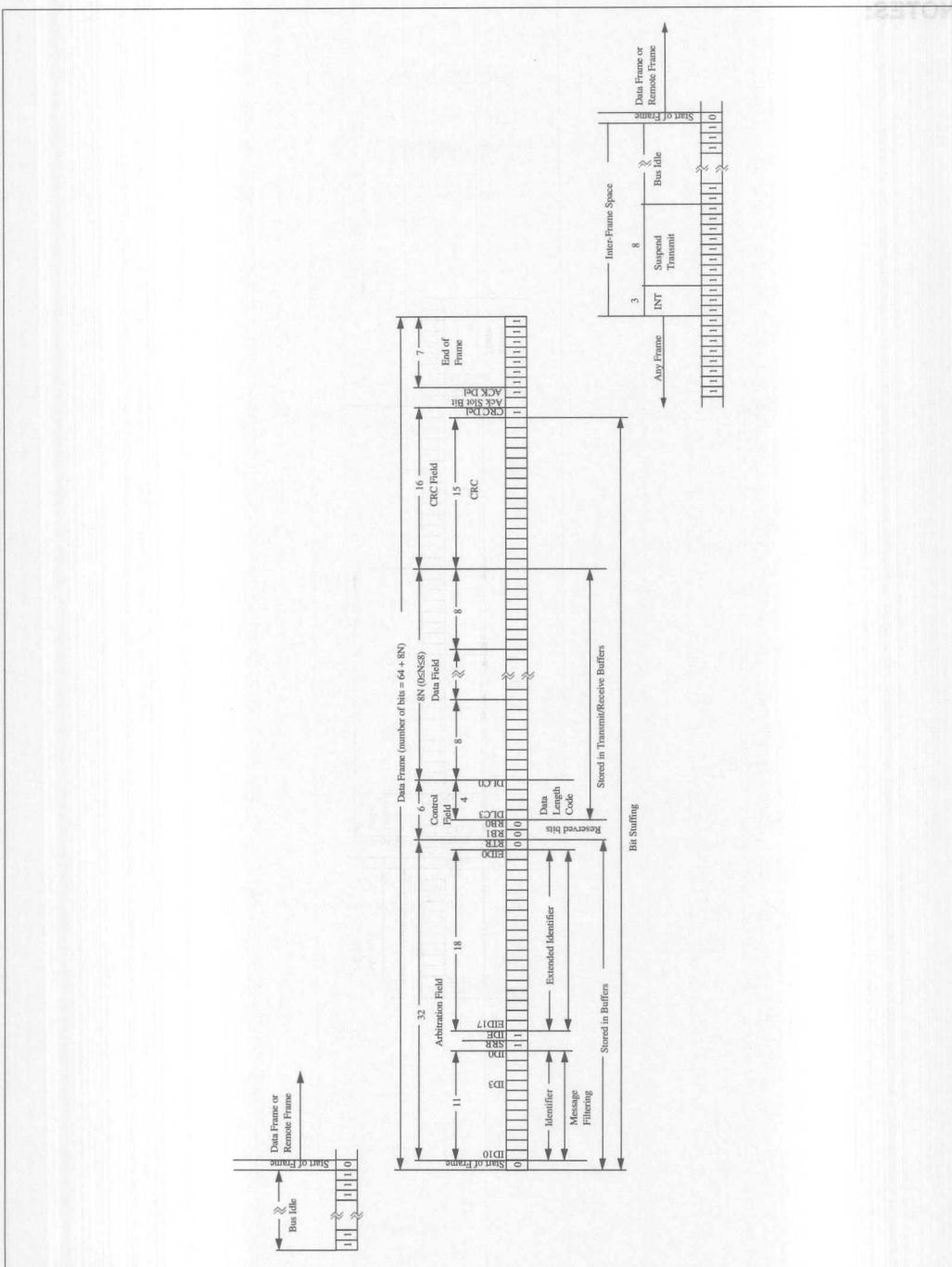
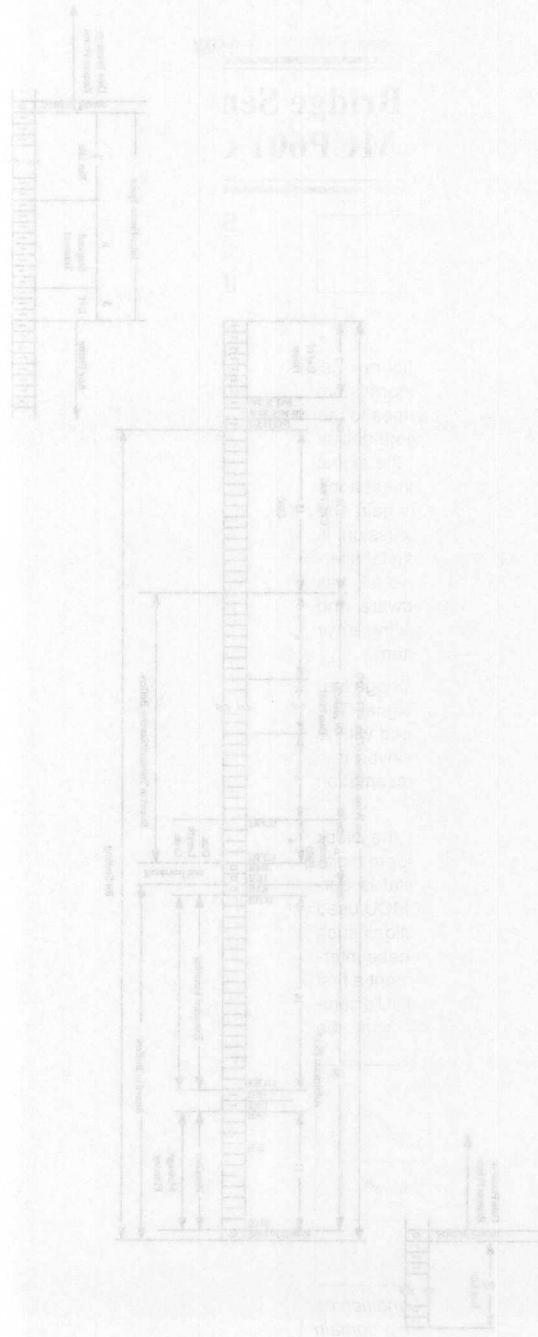


FIGURE 3: Extended Data Frame

AN713

NOTES:



Building a 10-bit Bridge Sensing Circuit using the PIC16C6XX and MCP601 Operational Amplifier

Author: *Bonnie C. Baker*
Microchip Technology Inc.

INTRODUCTION

Sensors that use Wheatstone bridge configurations, such as pressure sensors, load cells, or thermistors have a great deal of commonality when it comes to the signal conditioning circuit. This application note delves into the inner workings of the electronics of the signal conditioning path for sensors that use Wheatstone bridge configurations. Analog topics such as gain and filtering circuits will be explored. This discussion is complemented with digital issues such as digital filtering and digital manipulation techniques. Overall, this note's comprehensive investigation of hardware and firmware provides a practical solution including error correction in the data acquisition sensor system.

A sensor that is configured in a Wheatstone bridge typically supplies a low level, differential output signal. The application problem the designer is challenged with is to capture this small signal and eventually convert it to a digital format that gives an 8 to 12 bit representation of the signal.

The inexpensive design strategy shown in the block diagram in Figure 1 uses a low pass filter prior to digitization. The conversion from analog to digital is performed with the microcontroller (MCU). The MCU used in this circuit must have internal analog functions such as a voltage reference and a comparator. These internal analog building blocks are used to implement a first order modulator. This is combined with the MCU's computing power where a digital filter can be implemented.

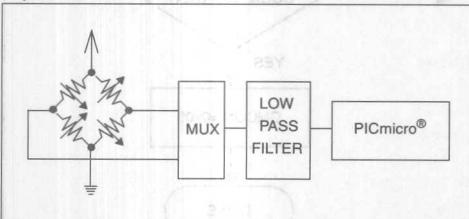


FIGURE 1: The bridge sensor signal conditioning chain filters high frequency noise in the analog domain then immediately digitizes it with a microcontroller.

BRIDGE SENSOR DATA ACQUISITION CIRCUIT IMPLEMENTATION

Figure 2 gives a detailed circuit for the block diagram shown in Figure 1. The analog portion of this circuit consists of the sensor (in this example a 1.2k Ω , 2mV/V load cell), an analog multiplexer, an amplifier and an R/C network. (A complete list of the load cell's specifications is given in Table 1.) An analog multiplexer is used to switch the two sensor outputs between a single ended signal path to the controller. The amplifier is configured as a buffer and used to isolate the sensor load from the R/C network. The R/C network implements the integrator function of a first order modulator. This network can also be used to adjust the input range to the MCU.

Rated Capacity	32 ounces (896 g)
Excitation	5V _{DC} to 12V _{DC}
Rated Output	2mV/V \pm 20%
Zero Balance	\pm 0.3mV/V
Operating Temperature	-55 to 95°C
Compensated Temperature	-5 to 50°C
Zero Balance over Temperature	0.036% FS/°C
Output over Temperature	0.036% FS/°C
Resistance	1200 Ω \pm 300 Ω
Safe Overload	150%
Full-Scale Deflection	0.01" to 0.05"

TABLE 1: Load Cell, LCL816G (Omega) Specifications.

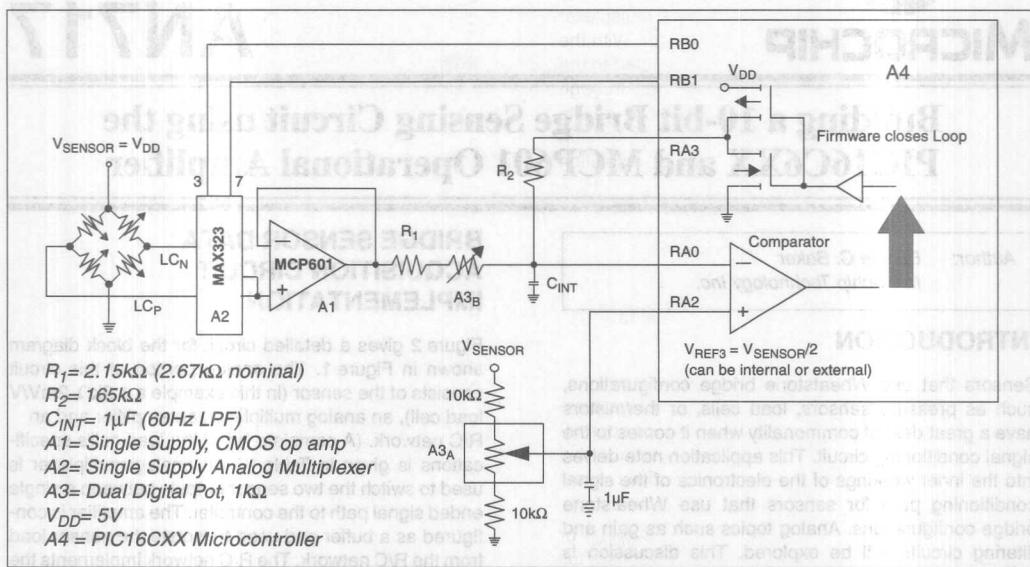


FIGURE 2: The combination of an R/C network and the microcontroller's analog peripherals can be used to perform an A/D conversion function.

In this circuit, the integrator function of the modulator is implemented with an external capacitor, C_{INT} . When RA3 of the PIC16CXX is set high, the voltage at RA0 increases in magnitude. This occurs until the output of the comparator (CMCON<6>) is triggered low. At this point, the driver to the RA3 output is switched from high to low. Once this has transpired, the voltage at the input to the comparator (RA0) decreases. This occurs until the comparator is tripped high. At this point, RA3 is set high and the cycle repeats. While the modulator section of this circuit is cycling, two counters are used to keep track of the time and of the number of ones versus zeros that occur at the output of the comparator. The firmware flow chart for this conversion process is shown in Figure 3.

DS00717A	
Rev. 0.01	Revised
08/01	08/01
Single Chip	Single Chip
Full-Ghost Detection	Full-Ghost Detection
Spec. Revision	Spec. Revision
DS00717A Rev. 0.01	DS00717A Rev. 0.01

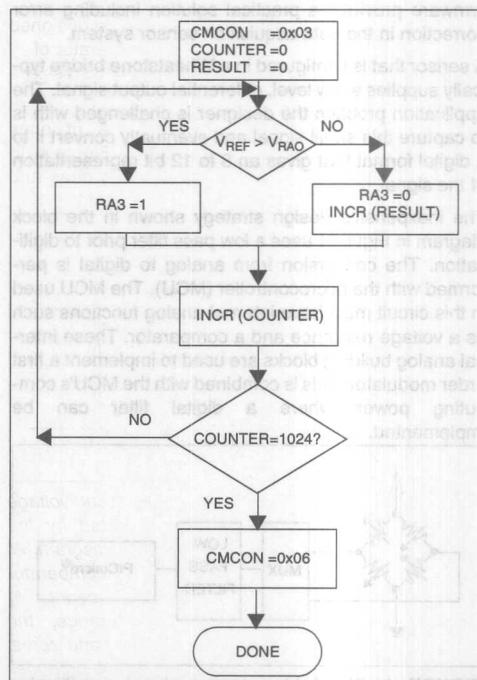


FIGURE 3: This microcontroller A/D conversion flow chart is implemented with the circuit shown in Figure 2. Care should be taken to make the time that every cycle takes through the flow chart constant.

Wheatstone bridge, the MCU switches the multiplexer (A2) to the leg of the other side of the sensor. With the voltage of the other leg of the sensor connected to the input of the amplifier, the controller cycles through another conversion for 1024 counts. The two results from these cycles are subtracted, giving the conversion results. This technique provides 10-bits of resolution with 9.9-bits of accuracy (rms).

The design equations for this circuit are:

$$V_{IN(CM)} = V_{REF3}$$

$$V_{IN(P\text{ to }P)} = V_{RA3}(P\text{ to }P)(R_1 / R_2)$$

with $V_{IN(CM)}$ approximates $V_{DD}/2$ or is equal to $(LC_P + LC_N)/2$, where:

V_{REF3} is the voltage reference applied to the comparator's non-inverting input and equal to approximately $V_{SENSOR}/2$. If made external, this reference voltage can be used to adjust offset errors.

$V_{IN(P\text{ to }P)}$ is equal to $(LC_P(\text{MAX}) - LC_N(\text{MIN}))$ or $(LC_P(\text{MIN}) - LC_N(\text{MAX}))$ which equals the sensor full scale range and $V_{RA3}(P\text{ to }P)$ is equal to $V_{RA3}(\text{MAX}) - V_{RA3}(\text{MIN})$ or approximately V_{DD} .

The system in this application note has been designed to have a full-scale input range to the comparator of $\pm 40.5\text{mV}$. Given 9.9-bit (rms) accuracy, the LSB size is $84.7\mu\text{V}$.

The transfer function of the percentage of ones counted versus input voltage is shown in Figure 4. In this diagram, both the duty cycle between ones and zeros as well as the pulse width is modulated.

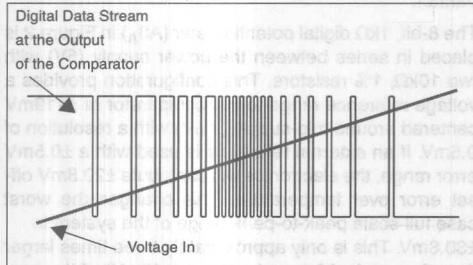


FIGURE 4: The relationship between input voltage and the number of ones that are counted by the controller is shown conceptually with this diagram. At low input voltages, the output of the comparator produces very few zeros within the 1024 counts. At voltages in the center of the input range, the comparator quickly toggles between ones and zeros. At higher voltages, the comparator output produces mostly zeros and very few ones.

SOLUTIONS

A wheatstone bridge is designed to give a differential output rendering a small voltage that changes proportionally to the sensor's excitation, i.e. pressure or temperature, etc. The dominant types of errors that a sensor exhibits in its transfer function can be categorized as offset, gain, linearity, noise, and thermal. These sensors also produce other errors such as hysteresis, repeatability, stability and aging that are beyond the scope of this application note. An equal contributor to the overall system errors is the offset, gain, and linearity errors from the active components in the signal conditioning path.

OFFSET ERRORS

The offset error of a system can be mathematically described with a constant additive to the entire transfer function as shown in Figure 5.

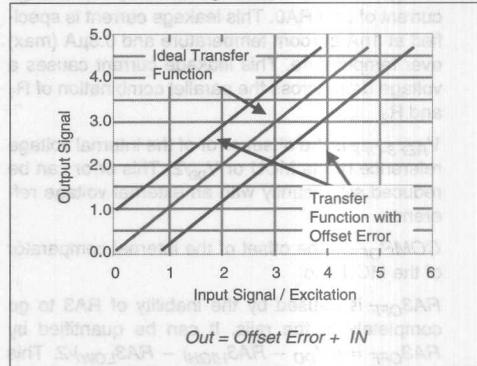


FIGURE 5: The offset error of a system can be described graphically with a transfer function that has shifted along the x-axis.

Typically, offset error is measured at a point where the input signal to the system is zero. This technique provides an output signal that is equal to the offset. This type of error can originate at the sensor or within the various components in the analog signal path. By definition, the offset error is repeatable and stable at a specified operating condition. If the operating conditions change, such as temperature, voltage excitation or current excitation, the offset error may also change.

The offset errors in this signal path come from the wheatstone bridge sensor, the operational amplifier (A1) offset, the port leakage current at RA0, the internal voltage reference (V_{REF3}) offset, the comparator offset, and the non-symmetrical output port of RA3. The only difference in the signal path of the two sensor outputs is the multiplexer channel, which interfaces directly with a high impedance CMOS operational amplifier. Otherwise, both sensor output signals are configured to travel down the same signal-conditioning path. Conse-

quently, the conversion data taken from the positive leg of the load cell sensor (LC_P) has the same offset and gain errors as the conversion data taken from the negative output of the load cell sensor (LP_N), with the exception of the bridge's offset error.

To accommodate these errors, the design equations for this circuit remains:

$$V_{IN(CM)} = V_{REF3}$$

$$V_{IN(P \text{ to } P)} = V_{RA3}(P \text{ to } P) (R_1 / R_2)$$

But, now the worst case variation of $V_{IN(P \text{ to } P)}$ is equal to $(LC_P(\text{MAX}) - LC_P(\text{MIN}) + LC_{OFF} + A1_{OFF} + RA0_{OFF} + V_{REF3-OFF} + COMP_{OFF} + RA3_{OFF})$

where:

ΔLC_{OFF} is the maximum offset voltage that can be generated by the load cell bridge

$A1_{OFF}$ is the offset voltage of the operational amplifier

$RA0_{OFF}$ is the offset error caused by the leakage current of port RA0. This leakage current is specified at 1nA at room temperature and 0.5 μ A (max) over temperature. This leakage current causes a voltage drop across the parallel combination of R_1 and R_2

$V_{REF3-OFF}$ is the offset error of the internal voltage reference of the MCU or $V_{DD}/2$. This error can be reduced significantly with an external voltage reference

$COMP_{OFF}$ is the offset of the internal comparator of the MCU and

$RA3_{OFF}$ is caused by the inability of RA3 to go completely to the rails. It can be quantified by $RA3_{OFF} = ((V_{DD} - RA3_{HIGH}) - RA3_{LOW})/2$. This formula assumes $V_{REF3} = V_{DD}/2$

The maximum magnitudes of these errors are summarized in Table 2.

Error Source	Offset Voltage over Temperature
Load Cell Bridge	$\pm 1.5\text{mV}$ in a 5V system
Op Amp	$\pm 2\text{mV}$
Port Leakage, RA0	$\pm 1.3\text{mV}$
Internal V_{REF}	$\pm 49\text{mV}$
Comparator	$\pm 10\text{mV}$
Output Port, RA3 (asymmetrical output swing)	5.5mV

TABLE 2: Maximum offset errors over temperature for the circuit shown in Figure 3.

Firmware Offset Calibration

The offset errors of the circuit can be calibrated in firmware. This is performed by subtracting the conversion code results of the positive leg of the sensor from the results of the negative leg of the sensor. The analog representation of the result of this calculation in firmware is:

$$V_{OUT} = LC_P + LC_{OFF} + A1_{OFF} + RA0_{OFF} + V_{REF3-OFF} + COMP_{OFF} + RA3_{OFF} - LC_N - A1_{OFF} - RA0_{OFF} - V_{REF3-OFF} - COMP_{OFF} - RA3_{OFF}$$

$$V_{OUT} = LC_P - LC_N + LC_{OFF}$$

This result illustrates the efficiency of using firmware to eliminate most of the offset errors, however, the trade-off for having offset adjustments performed by the MCU is dynamic range. In anticipation of these offset errors, the designer should increase the peak-to-peak analog input range of the conversion system. This will result in a conversion that has a wider dynamic range, consequently, lower accuracy.

To counteract this, the accuracy can be improved if more samples are taken in the conversion process. This technique will elongate the overall conversion time. Another technique that can be used is to perform a simple offset adjust in hardware which can be implemented with the digital potentiometer ($A3_A$).

Hardware Offset Calibration

Given the design equations for this circuit and the errors in Table 2, the total expected offset error over temperature for the electronics is $\pm 69.3\text{mV}$. With a sensor full-scale range of $\pm 10\text{mV}$, the dynamic range of the system would be ~ 7.9 times larger than the nominal, error free peak-to-peak range at the input of the comparator.

The 8-bit, 1k Ω digital potentiometer ($A3_A$) in Figure 2 is placed in series between the power supply (5V) with two 10k Ω , 1% resistors. This configuration provides a voltage reference range to the comparator of $\pm 119\text{mV}$ centered around mid-supply (2.5V) with a resolution of 0.5mV. If an external reference is used with a $\pm 0.5\text{mV}$ error range, the electronics will contribute $\pm 20.8\text{mV}$ offset error over temperature. This changes the worst case full-scale peak-to-peak range of the system to $\pm 30.8\text{mV}$. This is only approximately three times larger than the nominal full-scale output ($\pm 10\text{mV}$) of the sensor.

System Span (Gain) Errors

The span or gain of a system can be mathematically described as a constant, which is multiplied against the input signal. The magnitude of the span can easily be determined using the formula below:

$$\text{Ideal Output} = \text{Input} \times \text{Gain}$$

Span error is the deviation of the span multiple from ideal and can be described with the formula below:

$$\text{Actual Output} = \text{Input} \times \text{Span} (1 + \text{Span Error})$$

Examples of transfer functions with span error are shown in Figure 6. The plots in Figure 6 do not have offset errors.

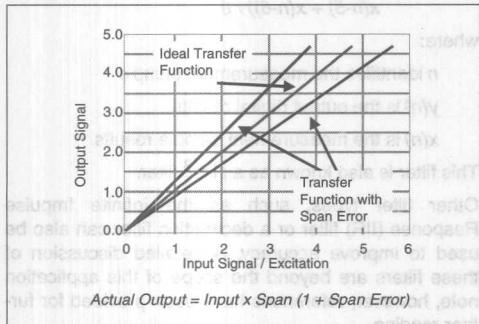


FIGURE 6: Span or gain errors can be described graphically as a transfer function that rotates around the intercept of the x and y-axis.

Firmware Span (Gain) Calibration

For this circuit, the span error of the sensor is less influential than the offset errors on the system. Span errors come from the Load Cell ($\pm 20\%$), the resistors ($\pm 1\%$), capacitor ($\pm 10\%$), and the ON resistance of the RA3 port (0.2%). This circuit can rely on firmware calibration with a reduction in the dynamic range of the system. The combination of the sensor error and the capacitor error increases the requirements on the input range to the modulator configuration.

Hardware Span (Gain) Calibration

Span errors can most effectively be removed in the analog domain. For instance, the span error of the sensor can be adjusted with the sensor's excitation voltage. As a trade-off for this adjustment strategy, the common mode voltage of the sensor is changed, creating offset errors with respect to the reference voltage (V_{REFS}) of the comparator. This problem can be alleviated by making the voltage reference ratiometric to the sensor excitation source. Span errors can also be adjusted with either R_1 or R_2 . In the circuit in Figure 2, the other half of the dual, 1kΩ digital potentiometer (A3B) is configured to perform this function. This type of adjustment does not change the offset error of the system. Finally, span errors can be corrected with changes to the integration capacitor. Of all of the span adjustments, this one is the most awkward to implement.

SYSTEM LINEARITY ERRORS

Linearity error differs from offset or span errors in that it has a unique affect on each individual code of the digitizing system. Linearity errors are defined as the deviation of the transfer function from a straight line. Some engineers define this error using a line that stretches between the end points of the transfer curve while others define it using a line that is calculated using a "best fit" algorithm. In either case, the linearity errors can cause significant errors in translating the sensor input (pressure, temperature, etc.) to digital code.

Linearity errors come in many forms as shown in Figure 7. Sometimes the linearity error of a system can be characterized with a multi-order polynomial, but more typically this error is difficult to predict from system to system, in which case, firmware piecewise linearization methods are usually used.

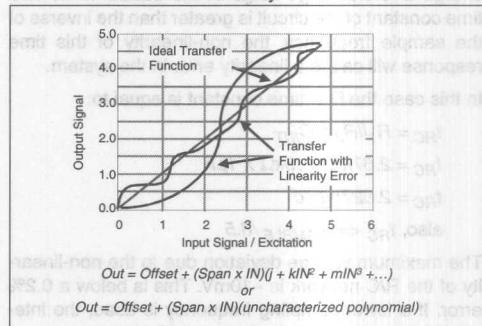


FIGURE 7: The linearity error of a sensor or system can sometimes be modeled and understood, which allows the designer to use predetermined algorithms in the MCU to minimize their affect. However, typically, these errors are not easily predicted and difficult to calibrate out of the signal path.

Linearity errors in this system originate primarily in the sensor and secondarily in the remainder of the signal conditioning circuit.

Firmware Linearization

Linearity errors can be calibrated out of the system in firmware using polynomial calculations if the transfer function is understood or piecewise linearization methods if the transfer function from part to part varies. If piecewise linearization is used, calibration data should be taken from the system and stored in EEPROM. Utilizing the calibration data, piecewise linearization is easily implemented using two 16 bit unsigned subtractions, one 16 bit unsigned multiplication and one 16 bit unsigned divide.

$$X_{CAL} = X_{FULL} / (Y_{FULL} - Y_{OFF}) \times (Y_{SAMP} - Y_{ZERO})$$

where:

X_{CAL} is equal to the calibrated results

X_{FULL} is the ideal full scale response saved in EEPROM

Y_{FULL} is the measured full scale response saved in EEPROM

Y_{OFF} is the measured offset error saved in EEPROM

Y_{SAMP} is the measured sample that requires linearization and calibration

This style of linearization correction also removes offset and span errors from the digital results.

Hardware Linearization

There are two components that generate linearity errors. The sensor can contribute up to $\pm 0.25\%$ FS error. The capacitor in this circuit can also contribute an appreciable error if care is not taken in limiting the charge and discharge range of the device. If the R/C time constant of the circuit is greater than the inverse of the sample frequency, the non-linearity of this time response will cause a linearity error in the system.

In this case the R/C time constant is equal to:

$$t_{RC} = R_1 / I R_2 \times C_{INT}$$

$$t_{RC} = 2.67k\Omega / 165k\Omega \times 1\mu F$$

$$t_{RC} = 2.627msec$$

also, $t_{RC} \leq t_{SAMPLE} / 6.5$

The maximum voltage deviation due to the non-linearity of the R/C network is $\sim 10mV$. This is below a 0.2% error. If a lower sampling frequency is used, the integrating capacitor must be increased in value.

SYSTEM NOISE ERRORS

Noise can plague the best of circuits, particularly circuits that have large analog segments. An effective way to approach noise problems is to use a basic list of guidelines in conjunction with a working knowledge of noise fundamentals. The checklist that every designer should have on hand includes:

1. Are bypass capacitors included in the design?
2. Is a low impedance ground plane implemented to minimize any ground noise across sensitive analog parts?
3. Are appropriate anti-aliasing filters used in front of the A/D converter?
4. Are the devices in the circuit too noisy?

Firmware Noise Reduction

The A/D converter described in this application note was modeled after a classic first order delta-sigma topology. In the digital domain, the data collection algorithm implements a simple average engine by default. This style of averaging, otherwise known as digital filtering, is also called a single order sinc filter or Finite Impulse Response (FIR) filter.

Further noise reduction algorithms can be implemented with the PIC MCU which will produce a system with higher accuracy. As an example, a third order FIR filter can be implemented with the following calculations in code:

$$y(n) = (2x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4) + x(n-5) + x(n-6)) / 8$$

where:

n identifies the measurement sample

$y(n)$ is the output digital results

$x(n)$ is the measurement sample results

This filter is also known as a sinc³ filter.

Other filter types, such as the Infinite Impulse Response (IIR) filter or a decimation filter can also be used to improve accuracy. A detailed discussion of these filters are beyond the scope of this application note, however, references have been provided for further reading.

Hardware Noise Reduction

In Figure 2, the R/C network that is used to implement the integrator function also serves as a low pass filter. This low pass filter is equal to:

$$f_{3dB} = 1 / (2\pi R_1 / I R_2 \times C_{INT})$$

Further noise reduction can be implemented by adding a second modulator stage at the input so this system. This implementation is shown in Figure 8.

Figure 8 shows a second modulator stage added to the system. This stage consists of a second R/C network that is connected to the input of the first modulator stage. The output of the second modulator stage is then fed into the first modulator stage.

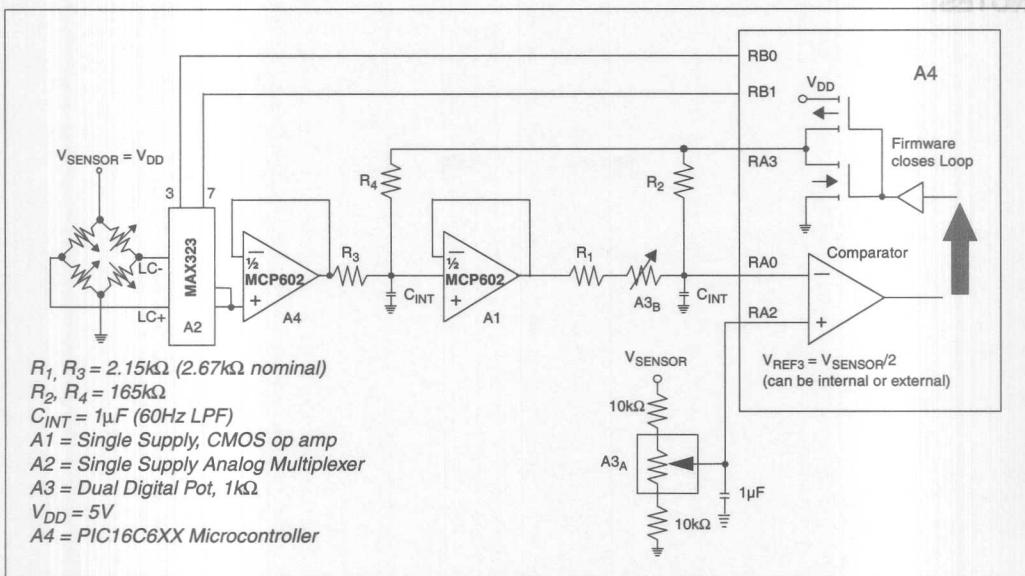


FIGURE 8: An additional modulator stages can reduce noise in the system even further. In this circuit, one modulator stage is added which improves the system from a 9.9-bit accurate (rms) system to an 11.1-bit accurate system.

REFERENCES

- Peter, Baker, Butler, Darmawaskita, "Making a Delta-Sigma Converter Using A Microcontroller's Analog Comparator Module", *AN700, Microchip Technology, Inc.*

Baker, Bonnie C., "Anti-aliasing, Analog Filters for Data Acquisition Systems", *AN699, Microchip Technology, Inc.*

Baker, Bonnie C., "Layout Tips for 12-Bit Converter Applications", *AN688, Microchip Technology, Inc.*

Morrison, Ralph, "Noise and Other Interfering Signals", *John Wiley & Sons, Inc. 1192*

Baker, Bonnie C., "Analog Circuit Noise Sources and Remedies", *EDTN Internet Magazine, Analog Avenue Tech Notes, Oct. 1998*

Baker, Bonnie C., "Noise Sources in Applications Using Capacitive Coupled Isolated Products", *AB-047, Burr-Brown Corp.*

Palacheria, Amar, "Implementing IIR Digital Filters", *AN540, Microchip Technology, Inc.*

Norsworthy, Schreier, Temes, "Delta-Sigma A/D Converters: Theory, Design, and Simulation", *IEEE Press*

NOTES:

Interfacing Microchip's MCP3201 Analog-to-Digital Converter to the PICmicro® Microcontroller

Author: Richard L. Fischer
Microchip Technology Inc.

INTRODUCTION

Many of the embedded control systems designed today require some flavor of a Analog-to-Digital (A/D) Converter. Embedded system applications such as data acquisition, sensor monitoring and instrumentation and Control all have varying A/D Converter requirements.

For the most part, these A/D Converter requirements are a combination of performance, cost, package size, and availability. Microchip offers a variety of solutions to meet these design requirements. The first possible solution is to implement the PICmicro® microcontroller (MCU). The PICmicro MCU offers many options for smart solutions. One of these features is the A/D Converter module. These A/D Converter modules are primarily successive approximation register (SAR) type and range in functionality from 8- to 12-bit with channel size ranges of 4 to 16. For example, the PIC16C77 has 8-channels of 8-bit A/D Converter, while the PIC17C766 has 16-channels of 10-bit A/D Converter. These on-board A/D Converter modules fit well into embedded applications, which requires a 10-35ksps A/D Converter.

For those applications which require a higher performance or remote sense capability, the Microchip MCP3201, 12-bit A/D Converter fits very nicely.

The MCP3201 employs a classic SAR architecture. The device uses an internal sample and hold capacitor to store the analog input while the conversion is taking place. Conversion rates of 100ksps are possible on the MCP3201. Minimum clock speed (10kHz or 625sp, assuming 16 clocks) is a function of the capacitors used for the sample and hold.

The MCP3201 has a single pseudo-differential input. The (IN-) input is limited to $\pm 100\text{mV}$. This can be used to cancel small noise signals present on both the (IN+) and (IN-) inputs. This provides a means of rejecting noise when the (IN-) input is used to sense a remote signal ground. The (IN+) input can range from the (IN-) input to V_{REF} .

The reference voltage for the MCP3201 is applied to V_{REF} pin. V_{REF} determines the analog input voltage range and the LSB size, i.e.:

$$\text{LSB size} = \frac{V_{\text{REF}}}{2^{12}}$$

As the reference input is reduced, the LSB size is reduced accordingly.

Communication with the MCP3201 is accomplished using a standard SPI™ compatible serial interface. This interface allows direct connection to the serial ports of MCUs and digital signal processors.

In order to simplify the design process for implementing the MCP3201, Microchip has written C and assembly code routines for a PIC16C67 to communicate with the MCP3201 A/D Converter.

Figure 1 shows the hardware schematic implemented in this application. Appendix A contains a listing of the C source code. Appendix B contains a listing of the assembly source code.

Wiring a 10-bit Analog-to-Digital Converter to a PICmicro™ MCU Interface

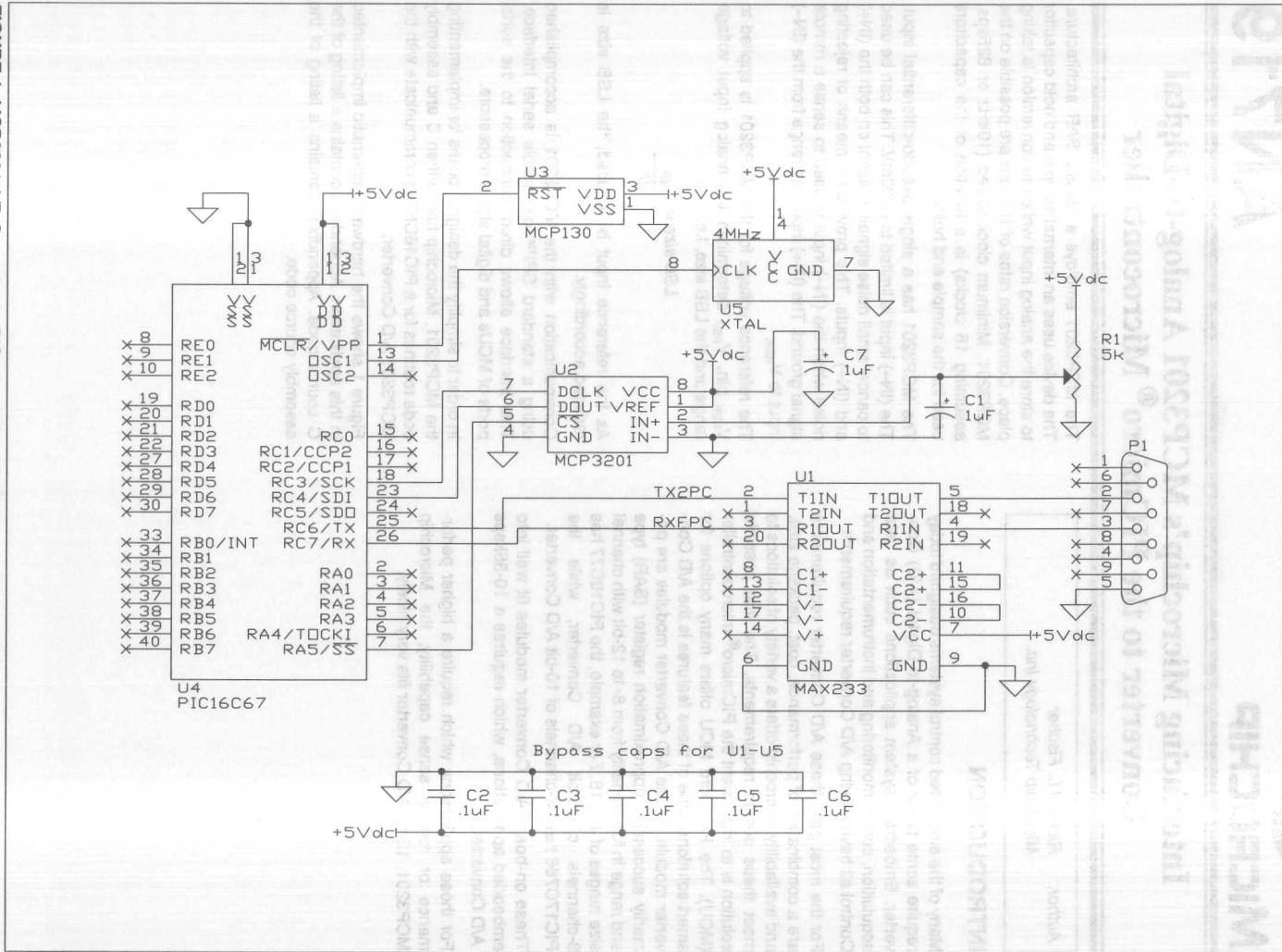


FIGURE 1: MCP3201 A/D Converter to PICmicro™ MCU Interface.

CIRCUIT DESCRIPTION

The serial interface of the Microchip MCP3201 A/D Converter has three wires, a serial clock input (DCLK), the serial data output (D_{OUT}) and the chip select input signal ($\overline{CS}/SHDN$). For this simple circuit interface, the PICmicro PIC16C67 SPI port is used. PortC:<3> is configured for the serial clock and PortC:<4> is the data input to the PICmicro. The SPI clock rate for this application is set at 1MHz.

The PIC16C67 is configured in the master mode with its CKP bit set to logic 1 and CKE bit set to logic 0. This configuration is the SPI bus mode 1,1.

A conversion is initiated with the high to low transition of $\overline{CS}/SHDN$ (active low). The chip select is generated by PORTA:<5> of the PICmicro. The device will sample the analog input from the rising edge on the first clock after \overline{CS} goes low for 1.5 clock cycles. On the falling edge of the second clock, the device will output a low null bit. The next 12 clocks will output the result of the conversion with the MSB first (See Figure 2 and Figure 3). Data is always output from the device on the falling edge of the clock. If the device continues to receive clocks while $\overline{CS}/SHDN$ is low, the device will output the conversion LSB first. If more clocks are provided to the device while $\overline{CS}/SHDN$ is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

As the analog input signal is applied to the IN+ and IN- inputs, it is ratioed to the V_{REF} input for conversion scaling.

$$\text{Digital output code} = \frac{V_{IN} \times F.S.}{V_{REF}}$$

Where:

V_{IN} = analog input voltage $V(IN+) - V(IN-)$

V_{REF} = reference voltage

F.S. = full scale = 4096

V_{REF} can be sourced directly from V_{DD} or can be supplied by an external reference. In either configuration, the V_{REF} source must be evaluated for noise contributions during the conversion. The voltage reference input, V_{REF} of the MCP3201 ranges from 250mV to 5V_{DC} which approximately translates to a corresponding LSB size from 61μV to 1.22mV per bit.

$$1.22mV = \frac{5V_{DC}}{2^{12} \text{ bits}}$$

For this simple application, the MCP3201 voltage reference input is tied to 5V_{DC}. This translates to a 1.22mV / bit resolution for the A/D Converter module. The voltage input to the MCP3201 is implemented with a multi-turn potentiometer. The output voltage range of this passive driver is approximately 0V_{DC} to 5V_{DC}.

Finally, a simple RS-232 interface is implemented using the USART peripheral of the PICmicro and a MAX233 transceiver IC. The USART transmits the captured A/D Converter binary value, both in ASCII and corresponding voltage to the PC terminal at 9600 baud.

With a few discrete components, a MCP3201 A/D Converter IC., and a PICDEM-2 demonstration board, this simple application can be implemented.

As with all applications which require moderate to high performance A/D Converter operation, proper grounding and layout techniques are essential in achieving optimal performance. Proper power supply decoupling and input signal and V_{REF} parameters must be considered for noise contributions.

SOURCE CODE DESCRIPTION

The code written for this application performs six functions:

1. PICmicro Initialization
2. A/D Conversion
3. Conversion to ASCII
4. Conversion to Decimal
5. Conversion to Voltage (*C code only)
6. Transmit ASCII, Decimal and Voltage to PC for display.

C CODE:

Upon power up, three initialization routines are called and executed. These routines initialize the PICmicro Port pins, USART peripheral and SSP module for SPI functionality. The default PICmicro SPI bus mode is 1,1. To place the PICmicro in SPI bus mode 0,0, comment out the "#define mode11" definition statement and rebuild the project.

Upon completion of the initialization routines, the main code loop is entered and executed every ~150ms. This continuous loop consists of performing an analog conversion, transmitting the results to the PC for display, delaying for ~150ms and then repeating the loop.

The A/D conversion sequence is initiated every time $\overline{CS}/SHDN$ is asserted. PortA:<5> is used as the $\overline{CS}/SHDN$ to the MCP3201. After asserting PortA:<5>, the SSPBUF register is written to, for initiating a SPI bus cycle. When the SPI cycle is complete, (BF flag is set to logic 1), the received data is read from the SSPBUF register and written to the RAM array variable "adc_databyte[1]". The SSPBUF register is again written to, which initiates a SPI bus cycle, and the second 8-bits are received and written to the RAM array variable "adc_databyte[0]". The $\overline{CS}/SHDN$ is then negated and the MCP3201 enters into the shutdown mode.

Next, the "Display_Adc_Result" routine is called and executed. Here the composite result, located in array variable "adc_databyte" is right adjusted one bit location. Then a printf statement is executed which formats

and sends the data to the USART for transmission to the PC for display. The data output is in three formats: ASCII, Decimal and Voltage.

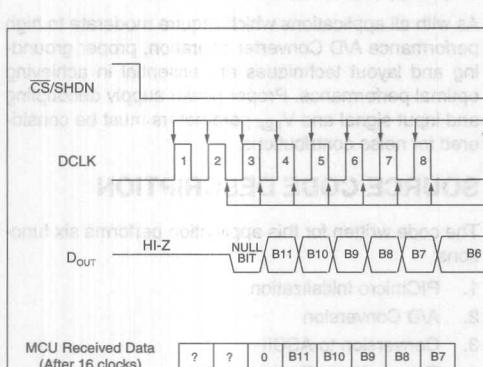


FIGURE 2: SPI Communication using 8-bit segments (Mode 0,0: DCLK idles low).

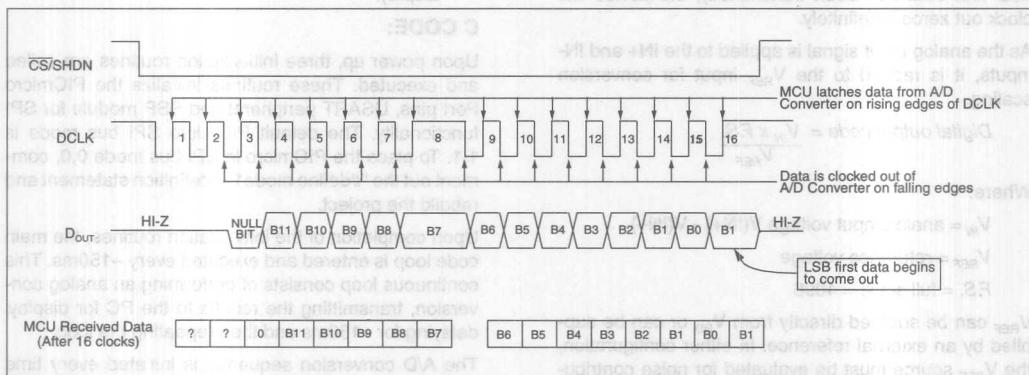


FIGURE 3: SPI Communication using 8-bit segments (Mode 1,1: DCLK idles high).

ASSEMBLY CODE:

Upon power up, three initialization routines are called and executed. These routines initialize the PICmicro Port pins, USART peripheral and SSP module for SPI functionality. The default PICmicro SPI bus mode is 1,1. To place the PICmicro in SPI bus mode 0,0, comment out the "#define mode11" statement and rebuild the project.

Upon completion of the initialization routines, the main code loop is entered and executed every ~150ms. This continuous loop consists of performing an analog conversion, converting the A/D Converter binary data into Decimal and ASCII and then transmitting the results to the PC for display, delaying for ~150ms and then repeating the loop.

The A/D conversion sequence is initiated every time CS/SHDN is asserted. PortA:<5> is used as the CS/SHDN to the MCP3201. After asserting PortA:<5>, the SSPBUF register is written to, for initiating a SPI bus cycle. When the SPI cycle is complete, (BF flag is set to logic 1), the received data is read from the SSPBUF register and written to the RAM variable "adc_result+1". The SSPBUF register is again written to, which initiates a SPI bus cycle, and the second 8-bits are received and written to the RAM variable "adc_result". Here the composite result, located in variable adc_result is right adjusted one bit location. The CS/SHDN is negated and the MCP3201 enters into the shutdown mode.

Next, the "Hex_Dec" and "Hex_Ascii" routines are executed which convert the raw A/D Converter binary data into Decimal and ASCII values. Then, the "Display_Data" routine is executed which sends the data to the USART for transmission to the PC for display.

REFERENCES

- Williams, Jim, "Analog Circuit Design", *Butterworth-Heinemann*.

Baker, Bonnie, "Layout Tips for 12-bit A/D Converter Applications", *AN688, Microchip Technology Inc.*

MCP3201 12-bit A/D Converter with SPI Serial Interface, *Microchip Technology, Document # DS21290B, 1999.*

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX A:

```

/*
 * Interfacing Microchip's MCP3201 ADC to the PICmicro MCU
 */

/*
 * Filename: mcp3201.c
 * Date: 06/30/99
 * File Version: 1.00
 * Compiler: Hi-Tech PIC C Compiler V7.84 PL1
 * MPLAB V4.12.00
 * Author: Richard L. Fischer
 * Microchip Technology Incorporated
 */

/*
 * Files required:
 *
 *      pic.h      - Hi-Tech provided file
 *      stdio.h    - Hi-Tech provided file
 *      config67.h
 *      mcp3201.h
 *
 *      mcp3201.c
 *      mprnt.c    - Hi-Tech provided file
 */

/*
 * This code demonstrates how the Microchip MCP3201 Analog-to-Digital
 * Converter (ADC) is interfaced to the Synchronous Serial Peripheral
 * (SSP) of the PICmicro MCU. For this application note the PICmicro
 * PIC16C67 is selected. The interface uses two Serial Peripheral
 * Interface (SPI) lines (SCK, SDI) on the PICmicro for the clock
 * (SCK) and data in (SDI). A chip select (CS) to the MCP3201 is
 * generated with a general purpose port line PORTA:<5>. The simple
 * application uses Mode 1,1 to define bus clock polarity and
 * phase.
 *
 * For this application, the SPI data rate is set to one fourth
 * (FOSC/4) of the microcontroller clock frequency. The PIC16C67
 * device clock frequency used for this application is 4MHz. This
 * translates to an ADC throughput of approximately 62.5kHz. In
 * order to obtain the maximum throughput (100kHz) from the
 * MCP3201 ADC the PIC16C67 should be clocked at 6.4Mhz.
 *
 */
#include <pic.h>                                // processor if/def file
#include <stdio.h>                               // configuration word definitions
#include "config67.h"
#include "mcp3201.h"

_CONFIG  ( CONBLANK & BODEN_ON & PWRTE_ON & CP_OFF & WDT_OFF & XT_OSC );

/* SPI Bus mode selection */
#define mode11                                     // comment out and rebuild for mode 00

```

```

***** MAIN PROGRAM BEGINS HERE *****

void main( void )
{
    Init_Ports();                                // initialize ports
    Init_SSP();                                  // initialize SSP module
    Init_Usart();                               // initialize USART module

    while ( TRUE )                                // loop forever
    {
        Read_AdC( );                            // initiate MCP3201 conversion and read result
        Display_AdC_Result();                  // display results via USART to PC
        Delay_10mS( 15 );                      // 150mS delay
    }
}

void Delay_10mS( char loop_count )
{
    unsigned int inner;                         // declare integer auto variable
    char outer;                                // declare char auto variable
                                                // stay in loop until done
    while ( loop_count )
    {
        for ( outer = 9; outer > 0; outer-- )
            for ( inner = 249; inner > 0; inner-- );
        loop_count--;
    }
}

void putch( char data )
{
    while ( !TRMT );
    TXREG = data;
}

void Read_AdC( void )
{
    CS = 0;                                     // assert MCP3201 chip select
    SSPBUF = 0x01;                               // initiate a SPI bus cycle
    while ( !STAT_BF );                          // wait until cycle completes
    adc.databyte[1] = SSPBUF;

    SSPBUF = 0x81;                               // initiate a SPI bus cycle
    while ( !STAT_BF );
    CS = 1;                                     // negate MCP3201 chip select
    adc.databyte[0] = SSPBUF;
}

void Display_AdC_Result( void )
{
    double temp;
    adc.result >>= 1;
    adc.result &= 0x0FFF;
    temp = ( adc.result * 0.001225585 );
}

```

AN719

```

printf( "Hex->0x%X : Decimal->%u : %4.3f Vdc\n\r", adc.result, adc.result, temp );
}

void Init_Usart( void )
{
    SPBRG = 25;           // set baud rate for 9600 @ 4MHz
    TXSTA = 0x24;          // BRGH = 1, enable transmitter
    RCSTA = 0x90;          // enable serial port
}

void Init_SSP( void )
{
    #ifdef model1
        SSPSTAT = 0b00000000;
        SSPCON = 0b00110000;
    #else if
        SSPSTAT = 0b01000000;
        SSPCON = 0b00100000;
    #endif
}

void Init_Ports( void )
{
    PORTA = 0b100000;
    PORTB = 0x00;
    PORTC = 0b11010000;
    PORTD = 0x00;
    PORTE = 0x00;

    TRISA = 0b000000;      // set PORTA data latches to initial state
    TRISB = 0x00;          // set PORTB data latches to initial state
    TRISC = 0b11010000;    // set PORTC data latches to initial state
    TRISD = 0x00;          // set PORTD data latches to initial state
    TRISE = 0x00;          // set PORTE data latches to initial state

    TRISA = 0b000000;      // set PORTA pin direction
    TRISB = 0x00;          // set PORTB pin direction
    TRISC = 0b11010000;    // set PORTC pin direction
    TRISD = 0x00;          // set PORTD pin direction
    TRISE = 0x00;          // set PORTE pin direction
}

```

```
/*
*   *
*   *   Filename:      mcp3201.h
*   *   Date:          06/30/99
*   *   File Version:  1.00
*   *
*   *
*****  

// FUNCTION PROTOTYPES DECLARED HERE  

void Read_Adc( void );
void Display_Adc_Result( void );
void Delay_10mS( char loop_count );
void Init_Usart( void );
void Init_SSP( void );
void Init_Ports( void );  

union {
    char databyte[2];
    unsigned int result;
} adc;  

#define TRUE      1
#define PortBit(port,bit)  ((unsigned)&(port)*8+(bit))
  

static bit CS @ PortBit(PORTA,5);           // MCP3201 Chip Select
```

AN719

```

/*
 *      *
 *      * Filename:      cnfig67.h
 *      * Date:          06/30/99
 *      * File Version:  1.00
 *      *
 *      *
 ****
 ****  CONFIGURATION BIT DEFINITIONS FOR PIC16C67 PICmicro  *****/
#define CONBLANK    0x3FFF

#define CP_ALL      0x00CF
#define CP_75       0x15DF
#define CP_50       0x2AEB
#define CP_OFF      0x3FFF
#define BODEN_ON    0x3FFF
#define BODEN_OFF   0x3FBF
#define PWRTE_OFF   0x3FFF
#define PWRTE_ON    0x3FF7
#define WDT_ON      0x3FFF
#define WDT_OFF     0x3FFB
#define LP_OSC      0x3FFC
#define XT_OSC      0x3FFD
#define HS_OSC      0x3FFE
#define RC_OSC      0x3FFF

```

Please check Microchip's Worldwide Website at www.microchip.com for the latest version of the source code.

APPENDIX B:

```

;*****
; Interfacing Microchip's MCP3201 ADC to the PICmicro MCU
;
;*****
; Filename: mcp3201.asm
; Date: 06/30/99
; File Version: 1.00
;
; Assembler: MPASM V2.30.00
; Linker: MPLINK V1.30.01
; MPLAB V4.12.00
;
; Author: Richard L. Fischer
; Company: Microchip Technology Incorporated
;
;***** Files required:
;
; mcp3201.asm
; hexdec.asm
; hexascii.asm
; p16c67.inc
; 16c67.lkr
;
;***** This code demonstrates how the Microchip MCP3201 Analog-to-Digital
; Converter (ADC) is interfaced to the Synchronous Serial Peripheral
; (SSP) of the PICmicro MCU. For this application note the PICmicro
; PIC16C67 is selected. The interface uses two Serial Peripheral
; Interface (SPI) lines (SCK, SDI) on the PICmicro for the clock
; (SCK) and data in (SDI). A chip select (CS) to the MCP3201 is
; generated with a general purpose port line PORTA:<5>. The simple
; application uses Mode 1,1 to define bus clock polarity and
; phase.
;
; For this application, the SPI data rate is set to one fourth
; (FOSC/4) of the microcontroller clock frequency. The PIC16C67
; device clock frequency used for this application is 4MHz. This
; translates to an ADC throughput of approximately 62.5kHz. In
; order to obtain the maximum throughput (100kHz) from the
; MCP3201 ADC the PIC16C67 should be clocked at 6.4Mhz.
;
;
;***** list p=16c67 ; list internal memory
; #include <p16c67.inc> ; processor specific variable definitions
; _CONFIG _BODEN_ON & _PWRTE_ON & _CP_OFF & _WDT_OFF & _XT_OSC TAP7464 ; configuration fuses
; #define mode11 ; if SPI bus mode 1,1 is desired
; ; else comment out and rebuild for mode 0,0
; ; if SPI bus mode 1,1 is desired
; ; else comment out and rebuild for mode 0,0
;
```

```

        .DATA      .NAME    .SIZE
adc_result    RES     2           ; variable used for context saving
offset       RES     1
temp         RES     1

        * This section contains end of DGA function arguments.  It is positioned
        * at the end of the code section to prevent it from being included in
        * the code size calculation.

TEMP_VAR1     UDATA_OVR   ; create udata overlay section
counthi      RES     1
countlo      RES     1

        GLOBAL  adc_result          ; make variables available to other modules
EXTERN   Hex_Dec             ; reference linkage
EXTERN   Hex_Ascii            ; reference linkage
EXTERN   adc_tempjh, adc_tempel ; reference linkage
EXTERN   thous                ; reference linkage

#define      CS      PORTA,5          ; MCP3201 Chip Select
#define      CR      0x0D             ; macro for carriage return
#define      LF      0x0A             ; macro for line feed

;*****
;***** RESET_VECTOR CODE 0x000 ; processor reset vector
        movlw    high start          ; move literal into W
        movwf    PCLATH              ; initialize PCLATH
        goto    start                ; go to beginning of program

;*****
;***** INT_VECTOR CODE 0x004 ; interrupt vector location
; no interrupt code needed for this application

;*****
;***** MAIN start CODE 0x040 ; set code section to start at 0x040
        call    Init_Ports           ; initialize ports
        call    Init_SSP              ; initialize SSP module
        call    Init_Usart             ; initialize USART module

;*****
;***** forever
        call    Read_Adc             ; read MCP3201 ADC
        call    Hex_Dec               ; convert adc_result to decimal
        call    Hex_Ascii              ; convert adc_result to ASCII
        call    Display_Data           ; display data to PC
        call    Delay_150MS            ; 150M delay
        goto    forever               ; continuos loop

;*****
;***** Read_Adc
        banksel  PORTA              ; linker to select SFR bank
        bcf     CS                   ; assert MCP3201 chip select
        movlw    0x01                 ; move literal into W
        banksel SSPBUF               ; linker to select SFR bank
        movwf    SSPBUF               ; initiate SPI bus cycle
        banksel SSPSTAT,BF           ; linker to select SFR bank
        btfss   SSPSTAT,BF           ; test, is bus cycle complete?
        goto    spi_busy1             ; wait, bus cycle not complete
        banksel SSPBUF               ; linker to select SFR bank
        movf    SSPBUF,WF             ; read SSPBUF and place into W
        banksel adc_result           ; linker to select GPR bank

```

```

movwf    adc_result+1  even      ; write SSPBUF to adc_result
        0x81    to bus not ready
banksel  SSPBUF 0x0000 to low   ; move literal into W
        SSPBUF 0x0000 to low   ; linker to select SFR bank
        SSPBUF 0x0000 to low   ; initiate SPI bus cycle
banksel  SSPSTAT 0x0000 to high  ; linker to select SFR bank
        SSPSTAT,BF test if ready
        btfss spi_busy2 set if not
        goto banksel 0x0000 to low
        banksel PORTA 0x0000 to low
        bsf    CS 0x0000 to high
        movf   SSPBUF,w read sspbuf
        banksel adc_result 0x0000 to low
        movwf  adc_result

        rrf    adc_result+1,f right
        rrf    adc_result,f right
        movlw  0x0F and adc_result
        andwf  adc_result+1,f result
        movf   adc_result,w result
        movwf  adc_temp1 set if ready
        movf   adc_result+1,w set if ready
        movwf  adc_temp2
        return

; Display ADC data ( ASCII and DECIMAL ) to USART
Display_Data
        banksel offset 0x0000 to low
        clrf   offset
        movlw  high msg1 0x0000 to low
        movwf  PCLATH at TSR if used
        incf   offset
        movwf  offset
        banksel offset 0x0000 to low
        clrf   offset
        movlw  msg1 0x0000 to low
        movwf  TXREG at TSR if used
        incf   offset
        banksel offset 0x0000 to low
        movwf  TXREG at TSR if used
        banksel offset 0x0000 to low
        btfss  TXSTA,TRMT
        goto   send_hex
        banksel TXREG 0 as TSR if used
        movwf  TXREG 0 as TSR if used
        incf   offset
        goto   txlp1

txlp1
        movf   offset,w
        call   msg1 0x0000 to low
        movwf  temp 0 as TSR if used
        btfsc  temp,7 0 as TSR if used
        goto   end_tx
        banksel TXREG 0 as TSR if used
        movwf  TXREG 0 as TSR if used
        banksel TXSTA,TRMT
        goto   $-1
        banksel offset
        incf   offset,f
        goto   txlp1

send_hex
        movlw  adc_temp1 0 as TSR if used
        movwf  FSR 0 as pointer
        movf   INDF,w
        banksel TXREG 0 as TSR if used
        movwf  TXREG 0 as TSR if used
        banksel TXSTA 0 as pointer
        btfss  TXSTA,TRMT 0 as TSR if used
        goto   $-1
        incf   FSR,f
        movlw  adc_temp1+4 0 as TSR if used
        subwf  FSR,w
        btfss  STATUS,C
        goto   send_hex1

send_hex1
        banksel offset 0x0000 to low
        clrf   offset
        movf   offset,w 0x0000 to low
        banksel offset 0x0000 to low
        btfss  offset,0x0000 to low
        goto   txlp2

```

```

call    msg2ba os_main_main ; retrieve table element
movwf  temp ; move element into temp
btfscl temp,7 ; test for end of string
goto   send_dec ; end of message so send the data
banksel TXREG and 162 ; linker to select SFR bank
movwf  TXREG ; initiate USART transmission
banksel TXSTA,TRMT or 162 ; linker to select SFR bank
btfscl TXSTA,TRMT or 162 ; test if TSR is empty
goto   $-1 ; stay in testing loop
banksel offset ; linker to select GPR bank
incf   offset,f ; update offset
goto   txlp2 ; increment table index
txlp2  doles os_main_main ; stay in transmit loop

; obtain variable address
send_dec movlw  thouz_lsb l_s2w_watbs ; initialize FSR as pointer
        movwf  FSR ; linker to select SFR bank
        movf   INDF,w ; retrieve data byte
banksel TXREG in thouz_lsb w_atbs ; linker to select SFR bank
        movwf  TXREG ; initiate USART transmission
banksel TXSTA,l_thouz_lsb w_atbs ; linker to select SFR bank
        btfscl TXSTA,TRMT or 162 ; test if TSR is empty
        goto   $-1 ; stay in loop
        incf   FSR,f ; update FSR
        movlw  thouz+4 ; compose end of string address value
        subwf  FSR,w ; do compare
        btfscl STATUS,C ; done with sending data
        goto   send_dec1 ; no, so send some more

        movlw  CR ; move literal into W
banksel TXREG doles os_main_main ; linker to select SFR bank
        movwf  TXREG ; initiate USART transmission
banksel TXSTA,do_acrd_main_watbs ; linker to select SFR bank
        btfscl TXSTA,TRMT ; test if TSR is empty
        goto   $-1 ; no, so stay in loop
        movlw  LF ; move literal into W
banksel TXREG doini_main_main ; linker to select SFR bank
        movwf  TXREG ; initiate USART transmission
banksel TXSTA,do_acrd_main_watbs ; linker to select SFR bank
        btfscl TXSTA,TRMT ; test if TSR is empty
        goto   $-1 ; no, so stay in loop
        return ; return from subroutine

; Delay for ~ 150ms
Delay_150ms
        movlw  D'150' ; move literal into W
banksel counthi_d150ms_main ; linker to select GPR bank
        movwf  counthi_d150ms_main ; initialize upper counter
outer   movlw  D'250' ; move literal into W
        movwf  countlo_d150ms_main ; initialize lower counter
inner   decf   countlo,f ; decrement counter low
        btfscl STATUS,Z_d150ms_main ; is result == 0
        goto   inner ; if NOT it goes to inner
        decf   counthi_d150ms_main ; else, decrement count high
        btfscl STATUS,Z_d150ms_main ; is result == 0
        goto   outer ; if YES goes to outer
        return ; return from subroutine

; Initialize USART Module
Init_Uart movlw  D'25' ; move literal into W
banksel SPBRG ; linker to select SFR bank
        movwf  SPBRG ; set baud rate for 9600 @ 4MHz
        movlw  B'00100100' ; move literal into W

```

```

        movwf    TXSTA          ; BRGH = 1, enable transmitter
        movlw    B'10010000'      ; move literal into W
        banksel  RCSTA          ; linker to select SFR bank
        movwf    RCSTA          ; enable serial port
        return   ; return from subroutine

; Initialize SSP Module
Init_SSP
#ifndef model1
        movlw    B'00110000'      ; move literal into W
        banksel  SSPCON          ; linker to select SFR bank
        movwf    SSPCON          ; enable Master SPI, bus mode 1,1, FOSC/4
        banksel  SSPSTAT         ; linker to select SFR bank
        clrf    SSPSTAT          ; Master sample data in middle, data xmt on
                                ; rising edge
#endif
#ifdef model0
        movlw    B'00100000'      ; move literal into W
        banksel  SSPCON          ; linker to select SFR bank
        movwf    SSPCON          ; enable Master SPI, bus mode 0,0, FOSC/4
        movlw    B'01000000'      ; move literal into W
        banksel  SSPSTAT          ; linker to select SFR bank
        movwf    SSPSTAT          ; Master sample data in middle, data xmt on
                                ; rising edge
#endif
#endif
        return   ; return from subroutine

; Initialize PORTS
Init_Ports movlw 0x00          ; move literal into W
        banksel PORTA          ; linker to select SFR bank
        movwf  PORTB          ; set PORTB data latches to initial state
        movwf  PORTD          ; set PORTD data latches to initial state
        movwf  PORTE          ; set PORTE data latches to initial state
        movlw  B'100000'        ; move literal into W
        movwf  PORTA          ; set PORTA data latches to initial state
        movlw  B'11010000       ; move literal into W
        movwf  PORTC          ; set PORTC data latches to initial state
        banksel TRISA          ; linker to select SFR bank
        clrf    TRISA          ; set PORTA pin direction
        clrf    TRISB          ; set PORTB pin direction
        clrf    TRISD          ; set PORTD pin direction
        clrf    TRISE          ; set PORTE pin direction
        movlw  B'11010000       ; move literal into W
        movwf  TRISC          ; set PORTC pin direction
        return  ; return from subroutine

TABLE_DATA CODE
msg1 addwf PCL,f           ; table starts here
DT    "HEX->.0x",80        ; generate computed goto
msg2 addwf PCL,f           ; generate computed goto
DT    "DECIMAL->,.80"      ; generate computed goto
END   ; directive 'end of program'

        (0000x0) 0000          ; 10x0
        (0000x1) 0001          ; 10x0
        (0000x2) 0002          ; 10x0
        (0000x3) 0003          ; 10x0
        (0000x4) 0004          ; 10x0
        (0000x5) 0005          ; 10x0
        (0000x6) 0006          ; 10x0
        (0000x7) 0007          ; 10x0
        (0000x8) 0008          ; 10x0
        (0000x9) 0009          ; 10x0
        (0000xA) 000A          ; 10x0
        (0000xB) 000B          ; 10x0
        (0000xC) 000C          ; 10x0
        (0000xD) 000D          ; 10x0
        (0000xE) 000E          ; 10x0
        (0000xF) 000F          ; 10x0

```

AN719


```

;*****  

;  

; Hex to ASCII conversion of ADC result for display  

;  

;*****  

;  

; File Name: hex2ascii.asm  

; Date: 06/30/99  

; File Version: 1.00  

;  

; Assembler: MPASM V2.30.00  

; Linker: MPLINK V1.30.01  

;          MPLAB V4.12.00  

;  

; Author: Richard L. Fischer  

; Company: Microchip Technology Incorporated  

;  

;*****  

;  

#include <p16c67.inc> ; processor specific variable definitions  

;  

GLOBAL Hex_Ascii ; make subroutine 'Hex_Ascii' available to  

GLOBAL adc_tempbh, adc_tempwl ; reference linkage  

;  

TEMP_VAR1 UDATA_OVR ; create udata overlay section  

adc_tempbh RES 2 ;  

adc_tempwl RES 2 ;  

;  

HEXASCII CODE ; create code section "HEXASCII"  

Hex_Ascii  

    banksel adc_tempbh  

    movf adc_tempwl,w  

    movwf adc_tempbh+1  

    movf adc_tempwl,w  

    movwf adc_tempbh+1  

    movlw 0x30  

    movwf adc_tempwl  

    swapf adc_tempwl,f  

    movlw 0x0F  

    andwf adc_tempwl,f  

    andwf adc_tempbh+1,f  

    movlw D'10'  

    subwf adc_tempwl,w  

    btfsc STATUS,C  

    goto add_37L  

    movlw 0x30  

    addwf adc_tempwl,f  

    movlw D'10'  

    subwf adc_tempbh+1,w  

    btfsc STATUS,C  

    goto add_37L  

    movlw 0x30  

    addwf adc_tempbh+1,f  

;  

chk_lsd movlw D'10'  

    subwf adc_tempbh+1,w  

    btfsc STATUS,C  

    goto add_37H  

    movlw 0x30  

    addwf adc_tempbh+1,f  

    goto exit  

;  

chk_msd movlw D'10'  

    subwf adc_tempbh+1,w  

    btfsc STATUS,C  

    goto add_37H  

    movlw 0x30  

    addwf adc_tempbh+1,f  

    goto exit

```

```
add_37L    movlw    0x37          ; move literal into W
            addwf    adc_temp1,f   ; compose ASCII character
            goto    chk_lsd       ; check least significant digit
add_37L1   movlw    0x37          ; move literal into W
            addwf    adc_temp1+1,f ; compose ASCII character
            goto    chk_msd       ; check most significant digit

add_37H    movlw    0x37          ; move literal into W
            addwf    adc_tempH+1,f ; compose ASCII character

exit      return               ; return from subroutine

END        ; directive 'end of program'
```

AN719

NOTES:

W orai fawdil evan :
notociaido II23A esqquid :
Della amollingia casei Meano :
W orai fawdil evan :
notociaido II23A esqquid :
n. w. amollingia casei Meano :

W orai fawdil evan :
notociaido II23A esqquid :

misuradre wail amoset :
notow fo bne' switwib :

Operational Amplifier Topologies and DC Specifications

Author: Bonnie C. Baker
Microchip Technology Inc.

INTRODUCTION

Operational amplifiers (op amps) are as prolific in analog circuits as salt and pepper is on food. They are sprinkled throughout the sensor data acquisition system, performing a variety of functions. For instance, at the sensor interface, amplifiers are used to buffer and gain the sensor output. The current or voltage excitation to the sensor, quite often is generated by an amplifier circuit. Following the front end sensor circuitry, an op amp is used to implement a low pass, band pass or high pass filter. In this portion of the circuit, gain stages are also implemented using programmable gain amplifiers or instrumentation amplifiers whose building blocks are the op amp. Analog-to-Digital (A/D) converters are most typically driven by an amplifier in order to achieve good converter performance.

Each one of these amplifier applications place unique demands on the device, so that one performance specification may be critical in one circuit, but not necessarily in another. This application note defines the DC

specifications of op amps and presents circuit applications where optimization of a particular specification is critical.

DEFINING THE OP AMP

Ideal Specifications

The op amp can be simply defined as an analog gain block with two signal inputs, two power supply connections and one output, as shown in Figure 1.

The input stage of the op amp has two terminals, the non-inverting (V_{IN+}) and inverting (V_{IN-}) inputs. For the ideal voltage feedback amplifier, both inputs are matched having no leakage current, infinite input impedance, infinite common mode rejection, zero noise and zero offset voltage (V_{OS}) between the terminals.

The power supply terminals (V_{DD} and V_{SS}) of the ideal op amp, have no minimum or maximum voltage restrictions. Additionally, the current from the power supply through the amplifier (I_{SUPPLY} , I_{DD} or I_o) is zero and any variation in the power supply voltage does not introduce errors into the analog signal path.

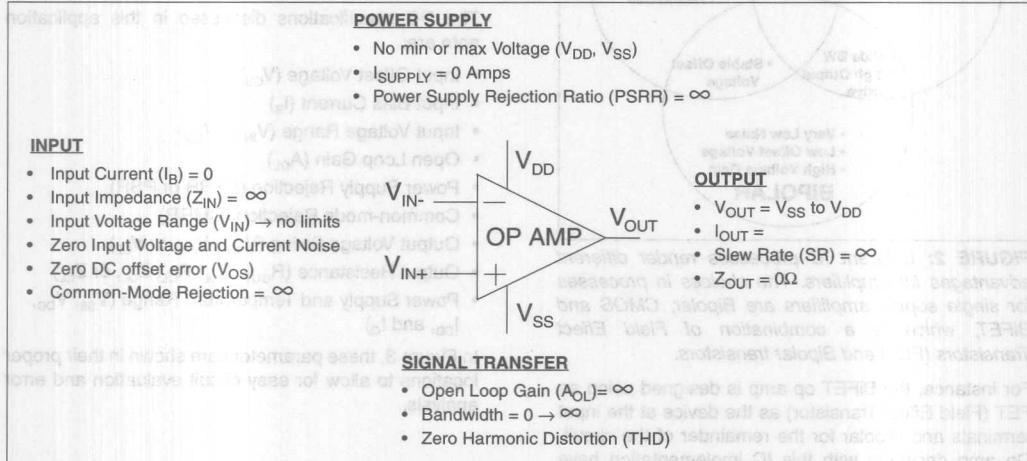


FIGURE 1: The ideal op amp description can be separated into four basic categories: input, power supply, output, and signal transfer.

In terms of the amplifier output, the swing capability equals or exceeds the voltage restrictions of the power supply. The output current (I_{OUT}) of this terminal can be infinite for indefinite periods of time, without causing reliability or catastrophic failures. The speed (SR) at which the output swings from rail to rail is instantaneous and the output impedance (Z_{OL} or Z_{CL}) is zero.

Finally, the open loop gain of the amplifier block is infinite and the bandwidth of the open loop gain is also infinite. To put the finishing touches on the signal transfer characteristics of the ideal amplifier, signals pass through the device without added distortion (THD) or noise.

Technology Limitations

This ideal amplifier does not exist. Consequently, performance specifications describe the amplifier so that the designer can assess the impact it will have on his circuit.

The errors that appear on the terminals of the op amp are a consequence of the semiconductor process and transistor implementation of the integrated circuit. In terms of the impact of the type of process that is used to design the amplifier, some generalities are summarized in Figure 2. These generalities are just that and not hard and fast rules.

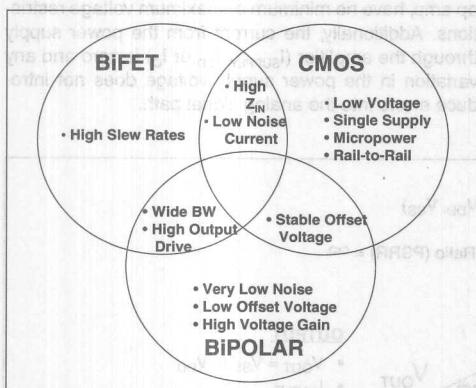


FIGURE 2: Different IC processes render different advantages for amplifiers. The choices in processes for single supply amplifiers are Bipolar, CMOS and BiFET, which is a combination of Field Effect Transistors (FET) and Bipolar transistors.

For instance, the BiFET op amp is designed using an FET (Field Effect Transistor) as the device at the input terminals and Bipolar for the remainder of the circuit. Op amp designed with this IC implementation have higher slew rates as compared to the pure Bipolar amplifier and CMOS amplifier.

In contrast, a pure Bipolar amplifier has NPN or PNP transistors at the input terminals. This allows the IC designer to achieve relatively low input offset voltage and voltage noise between the input terminals as well as higher open loop gains.

The commonality between the BiFET and Bipolar amplifiers are that they typically have wider bandwidths and higher output drive capability, as compared to the CMOS amplifier.

CMOS, on the other hand is well known for its low power, single supply op amps. The transistors in this style of amplifier are CMOS, allowing for an infinite input impedance and zero current leakage. This characteristic is similar in BiFET amplifiers. The degradation of these input impedances and leakage currents with the BiFET and CMOS input op amps are due to the required electrostatic discharge (ESD) cells that are added to the input terminals. CMOS amplifiers are also capable of rail-to-rail operation (in analog terms) while still having low quiescent current (current from the power supply).

The op amp specifications can be separated into two general categories, DC and AC. For the remainder of this application note, only the DC specifications will be discussed with accompanying detailed applications where that specification has an impact on the circuit performance. For discussions on AC specifications, refer to the application note from Microchip entitled "Operational Amplifier AC Specifications and Applications", AN723. (available December, 1999)

DC SPECIFICATIONS

The DC specifications discussed in this application note are:

- Input Offset Voltage (V_{OS})
- Input Bias Current (I_B)
- Input Voltage Range (V_{IN} or V_{CM})
- Open Loop Gain (A_{OL})
- Power Supply Rejection (PSRR or PSR)
- Common-mode Rejection (CMRR)
- Output Voltage Swing (V_{OUT} , V_{OH} , or V_{OL})
- Output Resistance (R_{OUT} , R_{OL} , R_{CL} , Z_{OL} , or Z_{CL})
- Power Supply and Temperature Range (V_{SS} , V_{DD} , I_{DD} , and I_O)

In Figure 3, these parameters are shown in their proper locations to allow for easy circuit evaluation and error analysis.

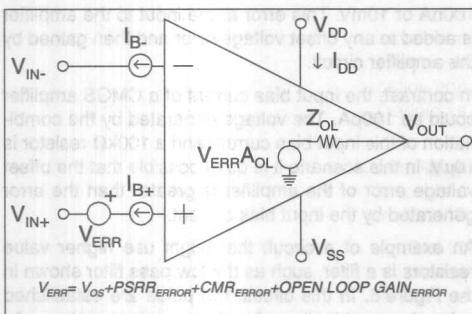


FIGURE 3: DC parameters for the op amp are modeled in a way to assist definition of specifications and easy error analysis of circuits.

For the remainder of this application note, these DC specifications will be defined and then evaluated within a sensitive application.

Input Offset Voltage (V_{OS})

Specification Discussion - The input offset voltage specification of an amplifier defines the maximum voltage difference that will occur between the two input terminals in a closed loop circuit while the amplifier is operating in its linear region. The input offset voltage is always specified at room temperature in terms of μ V or mV. The over temperature specification can be guaranteed as μ V/ $^{\circ}$ C as well as an absolute value of μ V or mV. Offset voltage is always modeled as a voltage source at the non-inverting input of the amplifier, as shown in Figure 3.

As with any amplifier specification, offset voltage can vary from part to part and with temperature, as shown in the distribution graphs in the Figure 4. The offset voltage of a particular amplifier does not vary unless the temperature, power supply voltage, common-mode voltage or output voltage changes, as shown in Figure 3 as part of V_{ERR} . The effects of these changes are discussed later.

As with any amplifier specification, offset voltage can vary from part to part and with temperature, as shown in the distribution graphs in the Figure 4. The offset voltage of a particular amplifier does not vary unless the temperature, power supply voltage, common-mode voltage or output voltage changes, as shown in Figure 3 as part of V_{ERR} . The effects of these changes are discussed later.

As with any amplifier specification, offset voltage can vary from part to part and with temperature, as shown in the distribution graphs in the Figure 4. The offset voltage of a particular amplifier does not vary unless the temperature, power supply voltage, common-mode voltage or output voltage changes, as shown in Figure 3 as part of V_{ERR} . The effects of these changes are discussed later.

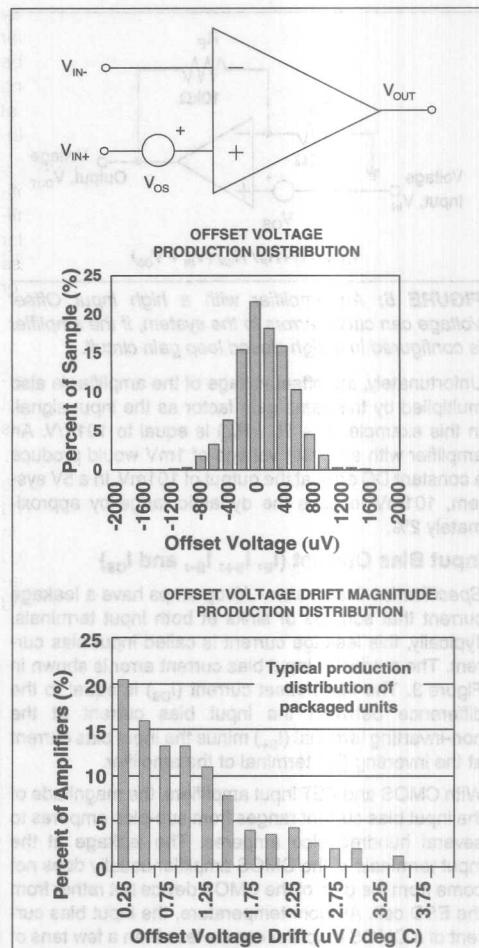


FIGURE 4: The input offset voltage of an amplifier varies from part to part but always falls within the stated specification voltage range.

Application Challenge - The offset voltage error of a particular amplifier may or may not be a problem, dependent on the application circuit. For instance, if a device is configured as a buffer (also known as a voltage follower), amplifiers with larger offset voltage errors, in the range of 2mV to 10mV, are usually not significantly different in performance than high precision amplifiers with extremely low offset voltage specifications, in the range of 100 μ V to 500 μ V. On the other hand, an amplifier with a high offset voltage that is in a high closed loop gain configuration can dramatically compromise the dynamic range of the circuit.

For example, the circuit in the Figure 5 is designed so that the analog input voltage (V_{IN}) is gained by:

$$V_{OUT} = (1 + R_F / R_{IN}) (V_{IN} + V_{OS})$$

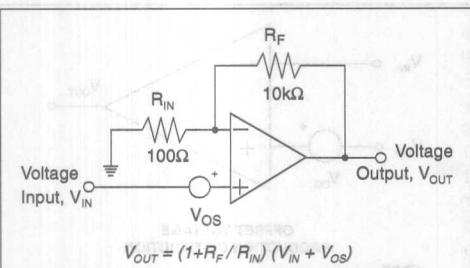


FIGURE 5: An amplifier with a high input Offset Voltage can cause errors in the system, if the amplifier is configured in a high closed loop gain circuit.

Unfortunately, the offset voltage of the amplifier is also multiplied by the same gain factor as the input signal. In this example, $(1 + R_F / R_{IN})$ is equal to 101V/V. An amplifier with an offset voltage of 1mV would produce a constant DC error at the output of 101mV. In a 5V system, 101mV lessens the dynamic range by approximately 2%.

Input Bias Current (I_B , I_{B+} , I_{B-} , and I_{os})

Specification Discussion - All op amps have a leakage current that sources or sinks at both input terminals. Typically, this leakage current is called input bias current. The model for input bias current error is shown in Figure 3. The input offset current (I_{os}) is equal to the difference between the input bias current at the non-inverting terminal (I_{B+}) minus the input bias current at the inverting (I_{B-}) terminal of the amplifier.

With CMOS and FET input amplifiers, the magnitude of the input bias current ranges from sub-pico amperes to several hundred pico amperes. The leakage at the input terminals of the CMOS amplifier usually does not come from the gate of the CMOS device but rather from the ESD cell. At room temperature, the input bias current of a CMOS amplifier can be less than a few tens of pico amperes. As the temperature increases, the ESD cells start to conduct current. This current appears at the input terminals of the amplifier.

In contrast, amplifiers with Bipolar inputs typically have input bias currents that range in the 10s of nano amps to several hundred nano amps. This current is the base current of the input Bipolar transistors. These amplifiers also have ESD cells, but the leakage from the base of the input transistor is much higher than the leakage from the ESD cells over temperature.

Application Challenge - Circuits that use high value resistors in the feedback loop or at the input of the amplifier are the most sensitive configurations for the op amp's input bias current error. For instance, if a high value resistor, such as 100kΩ is placed in series with the input of a Bipolar input amplifier that has an input bias current of 100nA, the resultant voltage is $100\text{k}\Omega \times$

100nA or 10mV. This error at the input to the amplifier is added to any offset voltage error and then gained by the amplifier circuit.

In contrast, the input bias current of a CMOS amplifier could be 100pA. The voltage generated by the combination of this input bias current and a 100kΩ resistor is 10μV. In this scenario it is quite possible that the offset voltage error of the amplifier is greater than the error generated by the input bias current.

An example of a circuit that might use higher value resistors is a filter, such as the low pass filter shown in the Figure 6. In this circuit, the poles are established using the combination of resistors and capacitors. As the cut-off frequency of a low pass filter is decreased, the RC time constants that generate the poles increase. In the situation where a low frequency, low pass filter is required, it is easy enough to find higher value capacitors. However, if board real-estate is an issue, higher value resistors are a more economical alternative.

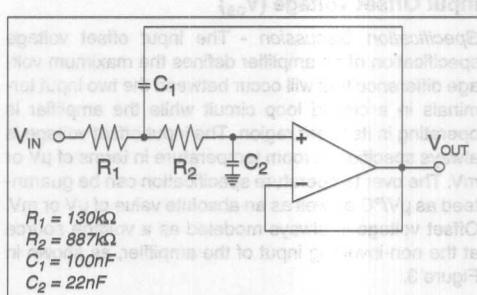


FIGURE 6: This Sallen-Key, 2nd order, 10Hz, Butterworth, low pass filter circuit has two large resistors in series with the non-inverting input of the op amp. Input bias current errors from a Bipolar op amp will cause a considerable amount of error. In contrast, the input bias current from CMOS or BiFET amplifiers will be low enough not to cause appreciable errors.

This RC relationship in combination with CMOS op amps can be used to an advantage with filters that have lower cut-off frequencies. Surface mount resistors can be found up to several mega ohms and surface mount film capacitors that are approximately the same size as the surface mount resistors can be found as high as several hundred nano farads. With this combination of passive devices, a compact, second order low pass filter can easily be designed down to 10Hz or lower.

In the example in Figure 6, a Bipolar amplifier with an input bias current of 100nA would generate a DC error through the resistor combination of R_1 and R_2 of 102.7mV. In contrast, a CMOS amplifier with an input bias current of 100pA would generate a DC error of 102.7μV.

Input Voltage Range (V_{IN} or V_{CM})

Specification Discussion - Each of the two input pins of the op amp has voltage swing restrictions. These restrictions are due to the input stage design. In the device product data sheet, the input voltage restrictions are clearly defined in one of two ways. Most commonly, the Input Voltage Range, V_{IN} , is specified as a separate line item in the specification table. This specification is also usually defined as a condition for the CMRR specification, input common-mode voltage range, V_{CM} . The more conservative specification of the two is where the Input voltage range is called out as a CMRR test condition because the CMRR test validates the input voltage range with a second specification.

The input voltage range is more a function of the input circuit topology rather than the silicon process. Although the input devices of the amplifier can be

CMOS, Bipolar or FET, there are three basic topologies that are used to design the input stage of single supply, voltage feedback amplifiers. These topologies are shown with a CMOS input stage in Figure 7. In Figure 7a, PMOS transistors (Q_1 and Q_2) are used for the first device at the input terminals. With this particular topology, the gate of both transistors can go 0.2 to 0.3V below the negative power supply voltage before these devices leave their active region. However, the input terminal can not go any higher than several hundred millivolts from the positive power supply voltage before the input devices are pulled out of their linear region. An amplifier designed with a PMOS input stage will typically have an input range of $V_{SS} - 0.2V$ to $V_{DD} + 1.2V$.

In Figure 7b, NMOS transistors (Q_1 and Q_2) are used for the first device at the input terminals. With this particular topology, the gate of both transistors can go 0.2 to 0.3V above the positive power supply voltage before these devices leave their active region. However, the input terminal can not go any lower than several hundred millivolts from the negative power supply voltage before the input devices are pulled out of their linear region. An amplifier designed with an NMOS input stage will typically have an input range of $-V_{SS} + 0.2V$ to $+V_{DD} - 1.2V$.

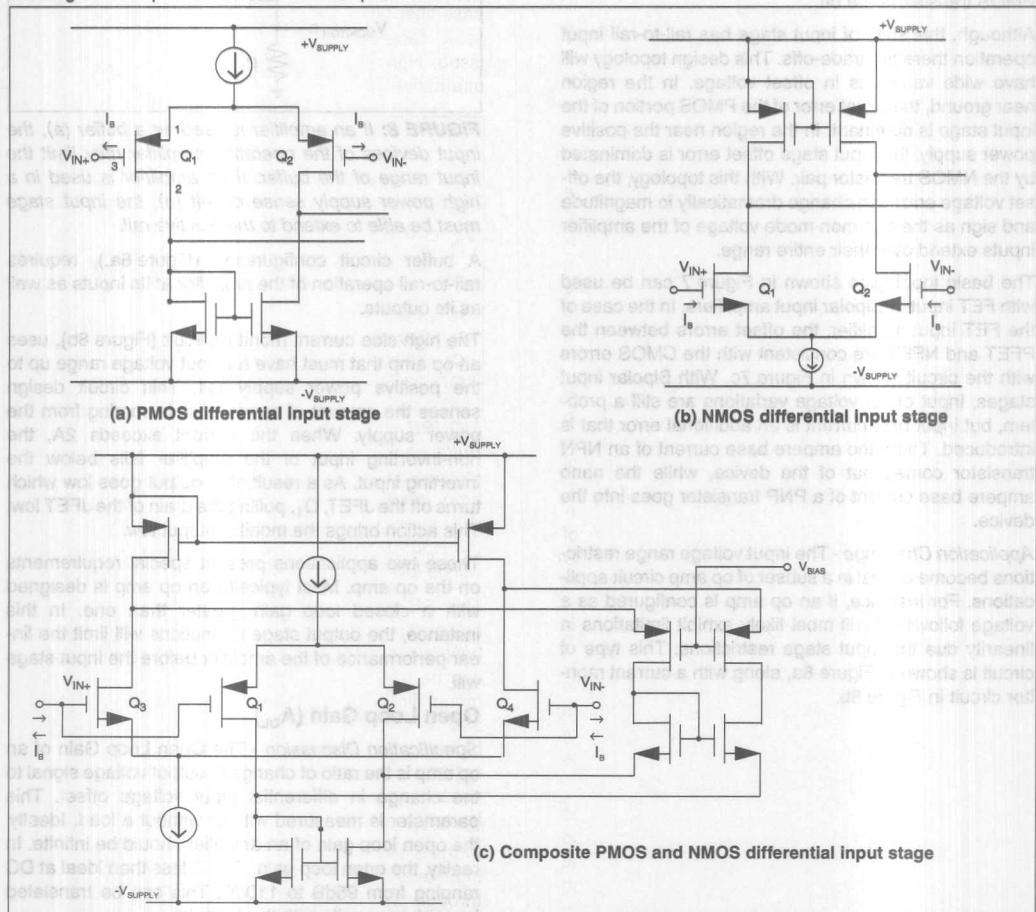


FIGURE 7: The input voltage range of an op amp is dependant on the topology of the input stage of the amplifier. The input stage can be constructed of PMOS (a) devices allowing for the input to swing below the negative supply or a NMOS differential pair (b) where the inputs can swing above the positive supply. A composite input stage (c) uses PMOS and NMOS differential pairs so the input voltage range can extend from above the positive rail to below the negative rail.

If the amplifier is designed with NMOS input transistors as shown in Figure 7b, the input range is restricted near the negative power supply voltage. In this case, the input terminals can be taken to a few tenths of a volt above the positive supply rail, but only to 1.2V above the negative supply rail.

If an amplifier input stage uses PMOS and NMOS transistors, it is configured as a composite stage, as shown in Figure 7c. With this topology, the amplifier effectively combines the advantages of the PMOS and NMOS transistors for true rail-to-rail input operation. When the input terminals of the amplifier are driven towards the negative rail, the PMOS transistors are turned completely on and the NMOS transistors are completely off. Conversely, when the input terminals are driven to the positive rail, the NMOS transistors are in use while the PMOS transistors are off.

Although, this style of input stage has rail-to-rail input operation there are trade-offs. This design topology will have wide variations in offset voltage. In the region near ground, the offset error of the PMOS portion of the input stage is dominant. In the region near the positive power supply, the input stage offset error is dominated by the NMOS transistor pair. With this topology, the offset voltage error can change dramatically in magnitude and sign as the common mode voltage of the amplifier inputs extend over their entire range.

The basic topologies shown in Figure 7 can be used with FET input or Bipolar input amplifiers. In the case of the FET input amplifier, the offset errors between the PFET and NFET are consistent with the CMOS errors with the circuit shown in Figure 7c. With Bipolar input stages, input offset voltage variations are still a problem, but input bias current is an additional error that is introduced. The nano ampere base current of an NPN transistor comes out of the device, while the nano ampere base current of a PNP transistor goes into the device.

Application Challenge - The input voltage range restrictions become critical in a subset of op amp circuit applications. For instance, if an op amp is configured as a voltage follower, it will most likely exhibit limitations in linearity due to the input stage restrictions. This type of circuit is shown in Figure 8a, along with a current monitor circuit in Figure 8b.

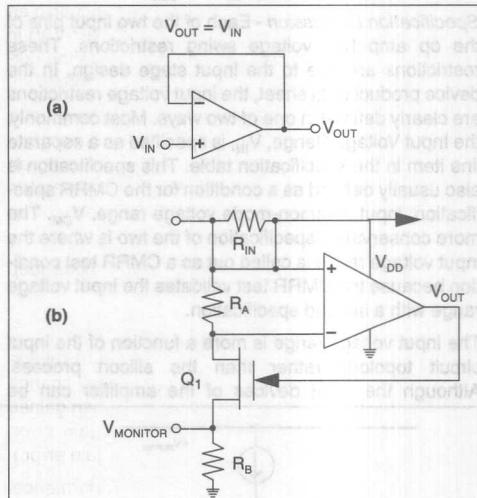


FIGURE 8: If an amplifier is used as a buffer (a), the input devices of the operation amplifier may limit the input range of the buffer. If an amplifier is used in a high power supply sense circuit (b), the input stage must be able to extend to the positive rail.

A buffer circuit configuration (Figure 8a.), requires rail-to-rail operation of the amplifier at its inputs as well as its outputs.

The high side current monitor circuit (Figure 8b), uses an op amp that must have an input voltage range up to the positive power supply rail. This circuit design senses the amount of current that is coming from the power supply. When the current exceeds 2A, the non-inverting input of the amplifier falls below the inverting input. As a result, that output goes low which turns off the JFET, Q_1 , pulling the drain of the JFET low. This action brings the monitor output low.

These two applications present special requirements on the op amp. Most typically, an op amp is designed with a closed loop gain greater than one. In this instance, the output stage restrictions will limit the linear performance of the amplifier before the input stage will.

Open Loop Gain (A_{OL})

Specification Discussion - The Open Loop Gain of an op amp is the ratio of change in output voltage signal to the change in differential input voltage offset. This parameter is measured with or without a load. Ideally, the open loop gain of an amplifier should be infinite. In reality, the open loop gain, A_{OL} , is less than ideal at DC ranging from 95dB to 110dB. This can be translated into volts per volts with the formula:

$$A_{OL} (V/V) = 10(A_{OL}(dB)/20)$$

DS00722A-page 2-256

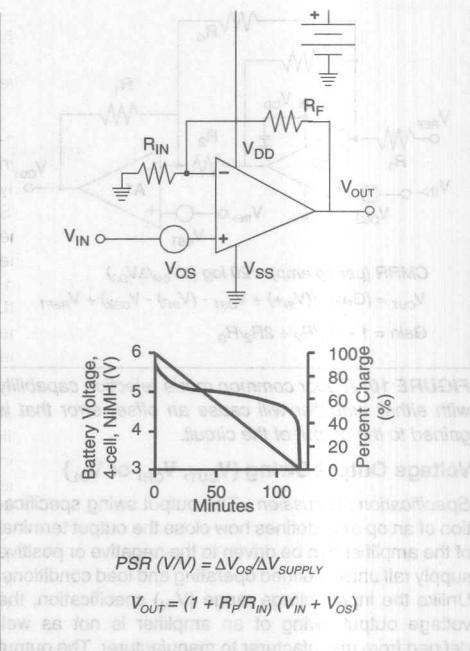


FIGURE 9: A battery powered application can see a change in power supply voltage of several hundreds of millivolts over the life of the product. If an op amp is configured with a high closed loop gain in these types of applications, it must have good DC power supply rejection.

Common Mode Rejection Ratio (CMRR)

Specification Discussion - The Common Mode Rejection Ratio of an amplifier describes the amplifier's input sensitivity to equivalent voltage changes of both inputs. This error manifests itself as an offset error ($CMRR_{ERROR}$), as shown in Figure 3.

$$CMRR(dB) = 20 \log (\Delta V_{CM} / \Delta V_{OS})$$

Where:

$$\Delta V_{OS} = CMRR_{ERROR}$$

Application Challenge - The specification range for CMRR in single supply amplifiers is from 45dB up to 90dB. Typically, this error becomes an issue when an amplifier is in a circuit where the input common mode voltage changes with input signal. A good example where this is the case, is when the amplifier is in a non-inverting configuration. A common circuit that has this configuration is shown in the Figure 10.

amplifier with an open loop gain of 100dB ($10^{10} V/V$) in an open loop configuration would be gained to the output of the amplifier to 1V.

In production runs, the open loop gain can vary up to 30% from part to part, consequently, a closed loop system is a more desirable configuration when using an amplifier, unless the amplifier is used as a comparator. With a closed loop system, the gain is dependent on the accuracy of the resistors in the circuit.

In a closed loop system, the effects of the open loop gain error is easily determined with:

$$A_{OL}(dB) = 20 \log (\Delta V_{OUT} / \Delta V_{OS})$$

This formula states that a change in the output voltage of the closed loop system will generate a small change in offset voltage. The offset voltage error is then gained by the closed loop system, generating a gain error. (Refer to Figure 3, where ΔV_{OS} = open loop gain error.)

A load will degrade the open loop gain performance. Some manufacturers recognize this and specify more than one test condition.

Power Supply Rejection (PSRR)

Specification Discussion - The power supply rejection ratio specification quantifies the amplifier's sensitivity to power supply changes. Ideally, the power supply rejection ratio should be infinite. Typical specifications for a power supply rejection ratio of an amplifier range from 60dB to 100dB.

As is with the open loop gain (A_{OL}) characteristics of an amplifier, DC and lower frequency power supply noise is rejected more than at higher frequencies. In a closed loop system, a less than ideal power supply rejection capability of an amplifier manifests itself as an offset voltage error as shown in Figure 3 ($PSRR_{ERROR} = \Delta V_{OS}$). This error is best described with the formula:

$$PSRR(dB) = 20 \log (\Delta V_{SUPPLY} / \Delta V_{OS})$$

The formula that describes power supply rejection is:

$$PSR(V/V) = \Delta V_{OS} / \Delta V_{SUPPLY}$$

Where:

$$V_{SUPPLY} = V_{DD} - V_{SS}$$

Application Challenge - An application where power supply rejection is critical is shown in Figure 9. In this circuit, a battery is used to power an amplifier which is configured in a high, closed loop gain of 101V/V. During the life of the battery, the output voltage ranges from 5.75V down to 4.75V. If the power supply rejection of the amplifier is 500µV/V (or 66dB), the error at the output of the amplifier over time would be 50.5mV. In a 12-bit system with a full-scale range of 4.096V, this would equate to a 50.5 counts worth of offset change over the life of the battery.

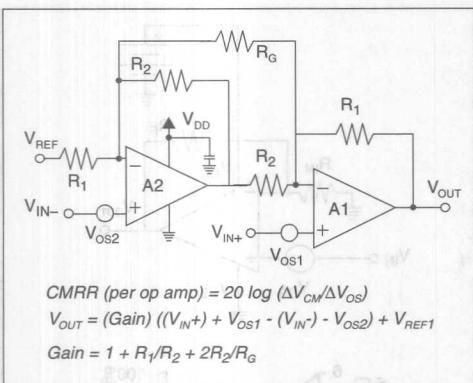


FIGURE 10: A poor common mode rejection capability with either amplifier will cause an offset error that is gained to the output of the circuit.

Voltage Output Swing (V_{OUT} , V_{OH} , or V_{OL})

Specification Discussion - The output swing specification of an op amp defines how close the output terminal of the amplifier can be driven to the negative or positive supply rail under defined operating and load conditions. Unlike the input voltage range (V_{IN}) specification, the voltage output swing of an amplifier is not as well defined from manufacturer to manufacturer. The output current as well as the amplifier's open loop gain (A_{OL}) are related to this specification. The output current is a test condition for the voltage output swing specification.

It is also a test condition for the open loop gain test, which validates the voltage output swing test with a second amplifier specification.

The output swing capability of the amplifier is dependent on the output stage design and the amount of current that the output stage is driving under test. With this portion of the specification, care should be taken when comparing amplifiers.

For instance, a single supply amplifier, MCP601, is used to generate the data in Table 1. It should be noted that the defined conditions of this specification have a significant influence on the amplifier's performance capability. All of these conditions, as well as others, can be found in op amp data sheets.

The key to comparing voltage output swing specifications, is to determine the amount of current that the amplifier is sinking or sourcing. The smaller the output current is, the closer the amplifier will swing to the rail.

If the load is specified as a current, this determination is easy. However, if the load is reference to $(V_{DD} - V_{SS})/2 + V_{SS}$, the output current is determined by dividing the voltage across the load resistor by the load resistor. It is useful to note that when the load is referenced to $(V_{DD} - V_{SS})/2 + V_{SS}$ the output of the amplifier will be sourcing or sinking half the current, as when the load is referenced to V_{DD} or V_{SS} .

The device in Table 1 was tested with the V_{DD} equal to 5V and V_{SS} equal to ground. Since this data was taken with one device, it does not necessarily represent the performance of all devices in the product family.

Output Voltage Swing	Test Conditions	Measured Output Swing from V_{SS} (mV)	Measured Output Swing from V_{DD} (mV)
High, to V_{DD}	w / 10kΩ load referenced to $(V_{DD} - V_{SS})/2 + V_{SS}$		11.2
High, to V_{DD}	w / 10kΩ load referenced to V_{SS}		20.4
High, to V_{DD}	w / 10kΩ load referenced to V_{DD}		1.95
High, to V_{DD}	w / amplifier source current equal to 100µA		3.8
Low, to V_{SS}	w / 10kΩ load referenced to $(V_{DD} - V_{SS})/2 + V_{SS}$	11.6	
Low, to V_{SS}	w / 10kΩ load referenced to V_{SS}	3.7	
Low, to V_{SS}	w / 10kΩ load referenced to V_{DD}	25.5	
Low, to V_{SS}	w / amplifier sink current equal to 100µA	8.1	

TABLE 1: This data was taken with one sample of the MCP601 op amp and demonstrates the effects of the output conditions on the output swing performance of that amplifier. This data was taken with no regard to the open loop gain of the amplifier.

The output voltage swing versus input offset voltage of this amplifier is shown in Figure 11. By using this plot, the open loop gain of the device can be calculated as the slope between two points. For example, the open loop gain of this amplifier using $V_{OUT} = 1V$ to $4V$, is $75dB$.

With this plot, it is noticeable that the linearity of the amplifier starts to degrade long before the output swing maximums are reached. If the output of an amplifier is operated beyond the linear region of this curve, the input to output relationship of the signal will be non-linear.

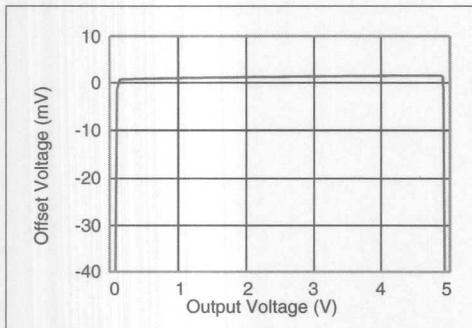


FIGURE 11: This graph shows the relationship between the output swing of an amplifier and input offset voltage with a $25k\Omega$ load and $V_{DD} = 5V$. The open loop gain of the amplifier can be calculated by selecting two points on the graph and calculating the slope. As the output swing of the amplifier goes towards the rail, the amplifier function eventually breaks down. This is first manifested with changes in input offset voltage.

Output Impedance (R_{OUT} , R_{CL} , R_{OL} , Z_{CL} , Z_{OL})

Specification Discussion - The fact that the output impedance of an op amp is low, makes the device useful in terms of "isolating" the impedance of two portions of a circuit. For this reason, low output impedance of an op amp is an important characteristic, but the precise output impedance is usually not specified.

When the output impedance is specified, it is given in terms of a resistance or impedance of a closed loop configuration (R_{CL} or Z_{CL}) or an open loop configuration (R_{OL} or Z_{OL}). Output impedance is most often specified as resistance.

Closed loop output resistance is the easiest to measure and is equal to:

$$R_{CL} = \Delta V_{OUT} / \Delta I_L$$

where

ΔV_{OUT} = the change in output voltage and

ΔI_L = the change in output current with a change in output voltage

The effective closed loop output impedance is less than the open loop output impedance by a factor equal to the reciprocal of the loop gain. The loop gain is equal to the open loop gain of the amplifier divided by the closed loop gain of the non-inverting circuit. For the circuit shown in Figure 12, the open loop loop resistance is equal to:

$$R_{OL} = R_{IN} / (A_{OL} / (1 + R_F / R_{IN}))$$

In this formula $(1 + R_F / R_{IN})$, is the non-inverting closed loop gain. This closed loop gain is also known as $1/\beta$.

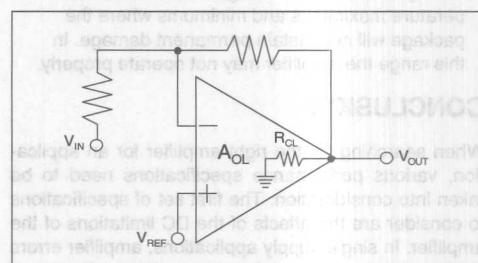


FIGURE 12: The closed loop output resistance of an amplifier is lessened by the magnitude of the open loop gain of the amplifier.

Power Supply Requirements (V_{SS} , V_{DD} , I_{DD} , I_Q)

Specification Discussion - Power supply voltage defines the acceptable difference between V_{DD} and V_{SS} which allows linear operation of the amplifier. If this voltage difference is less than specification, the amplifier may not operate reliably. If the power supply voltage is greater than specified, the amplifier most likely will operate as expected, but it is possible that damage may occur due to overvoltage stress on the internal transistors in the amplifier.

The power supply range is usually listed as a separate line item in the specification table in the product data sheet. Occasionally, the specification is called out as a condition under the PSRR specification.

Power supply current (I_{DD} or I_Q) is specified with no load. Typically, if a load is applied to the amplifier, a source current will primarily be pulled from V_{DD} , through the op amp output stage, and then through the load. A sink current will primarily result in an increase of V_{SS} .

Temperature Range

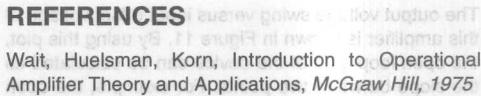
There are three types of temperature ranges that are specified with op amps.

- **Specified Temperature Range** - The range where the amplifier will meet specifications as called out in the specification table.
 - **Operating Temperature Range** - The range where the amplifier will operate without damage but performance is not necessarily guaranteed.
 - **Storage Temperature Range** - Defines the temperature maximums and minimums where the package will not sustain permanent damage. In this range the amplifier may not operate properly.

CONCLUSION

When searching for the right amplifier for an application, various performance specifications need to be taken into consideration. The first set of specifications to consider are the affects of the DC limitations of the amplifier. In single supply applications, amplifier errors such as input voltage swing, input offset voltage and input bias current could reduce the dynamic range of the amplifier. Conversely, in high gain circuits, the output voltage swing could cause signal clipping problems.

The second set of specifications to consider are the AC specifications. These issues are discussed in detail in the application note from Microchip entitled "Operational Amplifier AC Specifications and Applications", AN723 (available December, 1999).



"Operational Amplifier AC Specifications and Applications", Baker, Bonnie, *Microchip Technology, Inc.* AN723 (available December, 1999)



a no benig... m... nA - egned... netw...
MICROCHIP

DATA SHEET

AN723

Operational Amplifier AC Specifications and Applications

Author: *Bonnie C. Baker*
Microchip Technology Inc.

INTRODUCTION

This application note defines the ac specifications of voltage feedback operational amplifiers (Op Amps). Directly following these definitions, related amplifier circuits are given where the ramifications of the particular specifications causes less than optimum circuit performance. This is then followed with appropriate circuit solutions. The companion application note for DC specifications along with various application circuits is titled "Operational Amplifier Topologies and DC Specifications", AN-722 and available on Microchip's web site.

The performance specifications discussed in this application note are separated into the two categories listed below.

Frequency Domain Specifications

- Gain Bandwidth Product (GBWP)
- Open Loop Gain/Phase (A_{OL}, ϕ)
- Load Capacitance (C_L) - Output Impedance (Z_O)
- Full Power Bandwidth (FPBW)

Time Domain Specifications

- Slew Rate (SR)
- Settling Time (t_S)
- Overshoot

Topics such as bode plot generation, bode plot translation, stability analysis and feedback theory are discussed throughout this application note.

Additionally, there are numerous ac performance aspects of an operational amplifier that can be described from the frequency domain perspective or the time domain perspective. For instance, amplifier stability can be described in the frequency domain with the closed loop phase margin and its relationship to the amplifier open loop gain. In the time domain, the phase margin, in degrees, can be directly mapped to the settling time, and overshoot. Where appropriate, discussions in this application note will establish the correlation between these two domains.

The ac op amp open loop model that will be referred throughout this discussion is shown in Figure 1.

2
Application Notes

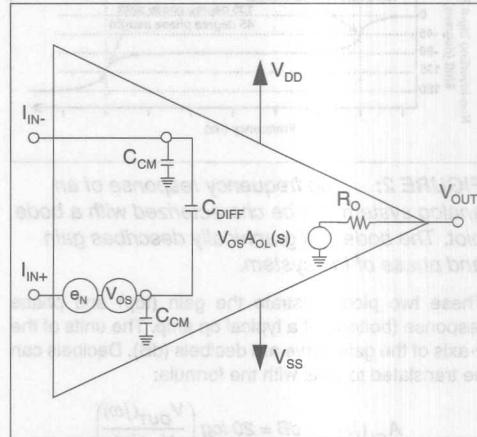


FIGURE 1: Voltage Feedback Operational Amplifier frequency model.

In this figure, the amplifier is shown with five terminals. The two input terminals have an offset voltage error (V_{OS}), a noise source (e_N) and differential capacitance (C_{DIFF}). The finite impedance of the output terminal is modeled using R_O . The amplifier's open loop gain over frequency is represented by the term $A_{OL}(j\omega)$. This gain term can be described in simple terms with the 2nd order transfer function:

$$A_{OL}(j\omega) = \frac{A_{OL}(DC)}{(as + 1)(bs + 1)}$$

Where:

a = the location of the dominant pole and

b = the location of the second pole

FREQUENCY DOMAIN SPECIFICATIONS

Bode Plot Analysis Method

The bode plot is a tool that is used to approximate the magnitude and phase of a transfer function. An example of the op amp gain and phase bode plots are shown in Figure 2.

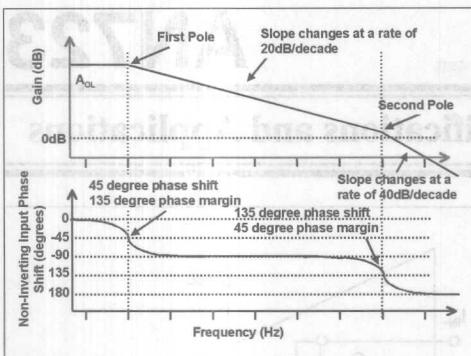


FIGURE 2: The frequency response of an analog system can be characterized with a bode plot. The bode plot graphically describes gain and phase of the system.

These two plots illustrate the gain (top) and phase response (bottom) of a typical op amp. The units of the y-axis of the gain curve are decibels (dB). Decibels can be translated to volts with the formula:

$$A_{OL}(j\omega) \text{ in dB} = 20 \log \left(\frac{V_{OUT}(j\omega)}{V_{IN}(j\omega)} \right)$$

The units of the y-axis of the phase plot is in degrees. Degrees can be converted to radians with the formula:

$$\text{Phase in radians} = (\text{Phase in degrees}) / 2\pi$$

Phase in degrees can be translated to phase delay or group delay (seconds), with the formula:

$$\text{Phase delay} = (\delta\text{phase in degrees}/\delta f)/360$$

Both plots are aligned on the same frequency scale with their respective x-axis.

Gain Bandwidth Product (GBWP)

Specification Discussion - The Gain Bandwidth Product (GBWP) of an amplifier is the product of the amplifier open loop gain times the frequency at any point in the frequency range where the amplifier response is attenuating at a rate of -20dB per decade of frequency.

By definition, if an amplifier is unity gain stable, it does not oscillate when the non-inverting input is used for the signal input and the inverting input is connected directly to the output. The unity gain bandwidth of the amplifier is equivalent to the amplifier's GBWP. The fact that an amplifier is unity gain stable implies that the phase shift from the non-inverting input to output is between zero and -180 degrees at the zero dB crossing of the open loop gain curve. Some amplifiers are not unity gain stable, in which case, the open loop gain zero crossing frequency is less than the GBWP.

Application Challenge - An amplifier configured as a buffer is shown in Figure 3. In this circuit, the buffer is used to electrically isolate conflicting impedances or to thermally isolate heavy loads. In this application, the amplifier selected must be unity gain stable.

A good stability test for a buffer is to apply a square wave to the input of the amplifier. The overshoot and ringing at the output of the amplifier will directly reflect the phase shift at the frequency where the gain is down 3dB.

For example, the bode plot of an amplifier that is not unity gain stable is shown in Figure 4. When this amplifier is configured as a buffer, the step response will show that the amplifier has a tendency to oscillate (Figure 5). The only remedy to this application problem is to select an amplifier that is unity gain stable. This is shown with the response of a second amplifier whose bode plot response is shown in Figure 6 and step response is shown in Figure 7.

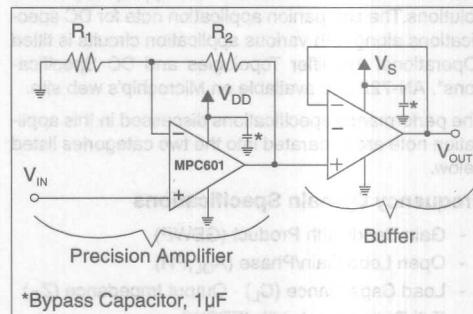


FIGURE 3: A typical application for an amplifier is the voltage buffer or follower.

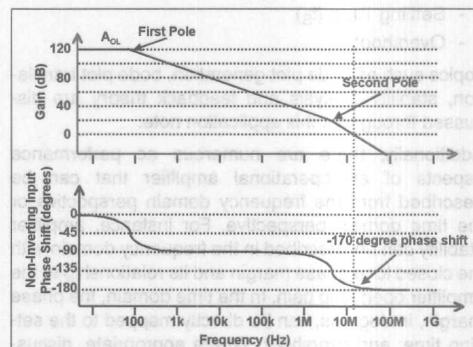


FIGURE 4: A bode plot of an amplifier that is not unity gain stable. This condition exists because the phase at the zero dB crossing frequency is nearly -180 degrees.

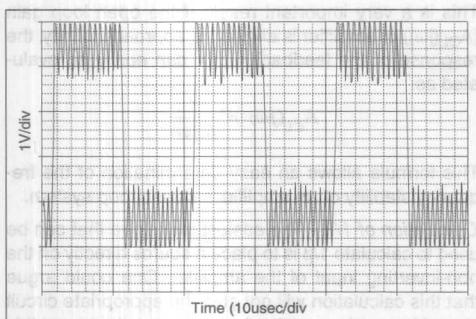


FIGURE 5: The unity gain step response of the amplifier specified with the bode plot of Figure 4 demonstrates the instability of the amplifier.

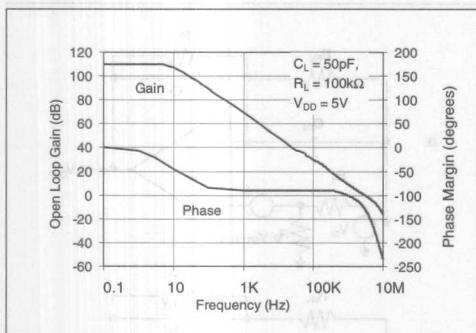


FIGURE 6: Bode plot of the MCP601 unity gain amplifier.

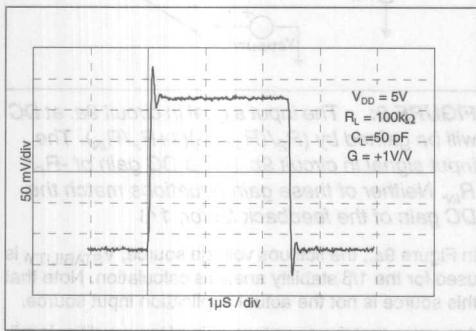


FIGURE 7: Step response of the unity gain stable MCP601 amplifier.

Open Loop Gain/Phase (A_{OL} , PH)

Specification Discussion - Ideally, the open loop gain of an amplifier is equal to the absolute value of the ratio of the voltage at the output terminal divided by the difference of the voltages applied between the two input terminals.

$$A_{OL}(dB) = 20 \log \left(\frac{V_{OUT}}{V_{IN+} - V_{IN-}} \right)$$

It would be ideal if the open loop gain ratio were infinite, but in reality, the complete frequency response of the open loop gain, $A_{OL}(j\omega)$, is less than ideal at DC and attenuates at a rate of 20dB/decade beginning at the frequency where the first pole in the transfer function appears. This is illustrated in the bode plot in Figure 2.

Usually, the first pole of the open loop response of an operational amplifier occurs between 1Hz to 10kHz. The second pole occurs at a higher frequency, nearer to the frequency where the open loop gain curve crosses 0dB. The gain response of an amplifier starts to fall off at 40dB/decade at the frequency where the second pole occurs.

The phase response of an amplifier in this open loop configuration is also fairly predictable. The phase shift or change from the non-inverting input to the output of the amplifier is zero degrees at DC. Conversely, the phase from the inverting input terminal to the output is equal to -180 degrees at DC.

At one decade (1/10 f_1) before the first pole, f_1 , the phase relationship of non-inverting input to output has already started to fall (~ -5.7 degrees). At the frequency where the first pole appears in the open loop gain curve (f_1), the phase has dropped to -45 degrees. The resultant phase continues to drop for another decade ($10f_1$) where it is 5.7 degrees above its final value of -90 degrees. This phase response discussion can be repeated for the second pole, f_2 .

What is important to understand is the ramifications of the changes in this phase relationship of the input to output of the amplifier. One frequency decade past the second pole, the phase shift of the non-inverting input is -180 degrees. At this same frequency, the phase shift of the inverting input to output is zero or -360 degrees. With this type of shift, V_{IN+} is actually inverting the signal to the output. In other words, the role of the two inputs have reversed.

Stability in Closed Loop Amplifier Systems

Typically, op amps are used with a feedback network in order to reduce the variability of the open loop gain response from part to part. A block diagram of this type of network is shown in Figure 8.

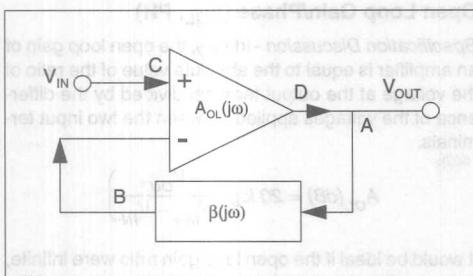


FIGURE 8: A block diagram of an amplifier circuit which includes the amplifier gain cell, A_{OL} , and the feedback network, β .

In Figure 8, β represents the feedback factor. Due to the fact that the open loop gain of the amplifier (A_{OL}) is relatively large, a fraction of the output voltage is fed back to the inverted input of the amplifier. If β were fed back to the non-inverting terminal, this small fraction of the output voltage would be added instead of subtracted. This configuration is appropriately called positive feedback and the output would eventually saturate.

The job of the feedback factor in this network is to reduce part-to-part variations in the system. However, if care is not taken, the feedback network can introduce unwanted frequency oscillations. This circumstance exists if the feedback factor is fed into the input of the system as a positive entity.

Closed Loop Transfer Function - The loop in Figure 8 can be analyzed by assuming an output voltage exists making the voltage at "A" equal to $V_{OUT}(j\omega)$. The signal passes through the feedback system, $\beta(j\omega)$, so that the voltage at "B" is equal to $\beta(j\omega)V_{OUT}(j\omega)$. This voltage is then summed to the input voltage which equates the voltage at "C". "C" is equal to $(V_{IN}(j\omega) - \beta(j\omega)V_{OUT}(j\omega))$. With the signal passing through the gain cell, $A_{OL}(j\omega)$, the voltage at point "D" is equal to $A_{OL}(j\omega)(V_{IN}(j\omega) - \beta(j\omega)V_{OUT}(j\omega))$. It should be noted that this voltage is equal to the original node, "A", or V_{OUT} . The formula that describes this complete closed loop system is equal to:

$$\frac{V_{OUT}(j\omega)}{V_{IN}(j\omega)} = \frac{A_{OL}(j\omega)}{(1 + A_{OL}(j\omega)\beta(j\omega))}$$

By collecting the terms, the manipulated transfer function becomes:

$$\frac{V_{OUT}(j\omega)}{V_{IN}(j\omega)} = \frac{A_{OL}(j\omega)}{(1 + A_{OL}(j\omega)\beta(j\omega))}$$

This formula is essentially equal to the closed loop gain of the system, or $A_{CL}(j\omega)$.

This is a very important result. If the open loop gain ($A_{OL}(j\omega)$) of amplifier is allowed to approach infinity, the response of the feedback factor can easily be evaluated as:

$$A_{CL}(j\omega) = \frac{1}{\beta(j\omega)}$$

This formula allows an easy determination of the frequency stability of an amplifier's closed loop system.

Calculation of $1/\beta$ - The easiest technique that can be used to calculate $1/\beta$ is to place a source directly on the non-inverting input of the amplifier. One could argue that this calculation will not give the appropriate circuit closed loop gain equation for the actual signal and this is true. But this calculation can be used to ascertain the level of circuit stability.

The circuits in Figure 9 are used to show how to calculate $1/\beta$.

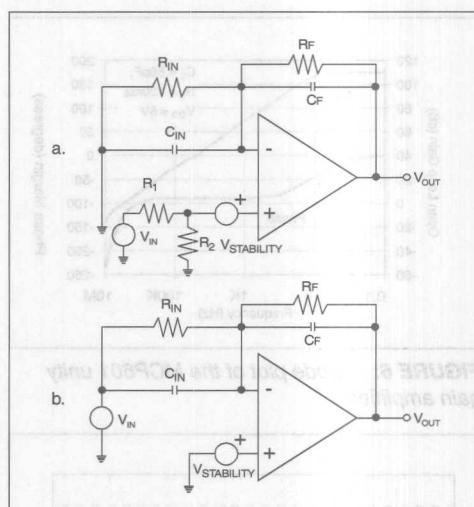


FIGURE 9: The input signal in circuit 9a. at DC will be gained by $R_2/(R_1+R_2)(1+R_F/R_{IN})$. The input signal in circuit 9b. has a DC gain of $-R_F/R_{IN}$. Neither of these gain equations match the DC gain of the feedback factor, $1/\beta$.

In Figure 9a., the fictitious voltage source, $V_{STABILITY}$, is used for the $1/\beta$ stability analysis calculation. Note that this source is not the actual application input source.

Assuming that the open loop gain of the amplifier is infinite, the transfer function of this circuit is equal to:

$$\frac{V_{OUT}}{V_{STABILITY}} = \frac{1}{\beta}$$

$$\frac{1}{\beta} = 1 + \frac{R_F//C_F}{R_{IN}//C_{IN}}$$

In this equation above, when ω is equal to zero:

$$\frac{1}{\beta(j\omega)} = \frac{(R_{IN}(j\omega)R_F C_F + 1) + R_F((j\omega)R_{IN}C_{IN} + 1)}{R_{IN}((j\omega)R_F C_F + 1)}$$

As ω approaches infinity,

$$\frac{1}{\beta(j\omega)} = 1 + \frac{C_{IN}}{C_F}$$

The transfer function has one zero and one pole. The zero is located at:

$$f_z = \frac{1}{(2\pi R_{IN} \parallel R_F(C_{IN} + C_F))}$$

$$f_p = \frac{1}{2\pi R_F C_F}$$

The bode plot for the $1/\beta(j\omega)$ transfer function of the circuit in Figure 9a. is shown in Figure 10.

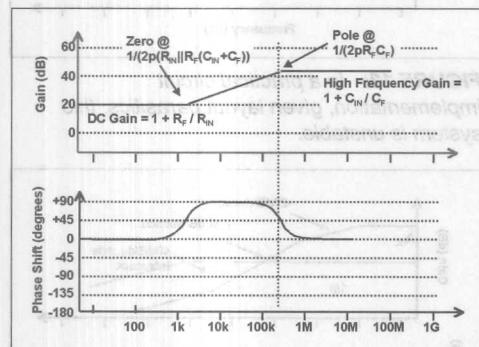


FIGURE 10: These are the bode plots of the inverse of the feedback factor ($1/\beta$) for circuit in Figure 9a. using $V_{STABILITY}$ as the input source.

Once again, in Figure 9b, the input source that is used for this analysis is not the same as the input source for the actual application circuit. However, the amplifier stability is determined in the same manner. The closed loop transfer function, using $V_{STABILITY}$ is equal to:

$$\frac{V_{OUT}}{V_{STABILITY}} = \frac{1}{\beta}$$

$$\frac{1}{\beta} = 1 + \frac{R_F \parallel C_F}{R_{IN} \parallel C_{IN}}$$

$$\frac{1}{\beta(j\omega)} = \frac{(R_{IN}(j\omega)R_F C_F + 1) + R_F((j\omega)R_{IN}C_{IN} + 1)}{R_{IN}((j\omega)R_F C_F + 1)}$$

Note that the transfer functions of $1/\beta$ between Figure 9a. and Figure 9b. are identical.

Determining System Stability - In a closed loop amplifier system, stability can be determined if the phase margin of the system is known. In this analysis, the Bode stability analysis technique is commonly used. With this technique, the magnitude (in dB) and phase response of both the open loop response of the amplifier and circuit feedback factor are included in a Bode plot.

The system closed loop gain is equal to the lesser (in magnitude) of the two gains. The phase response of the system is the equal to the open loop gain phase shift minus the inverted feedback factor's phase shift.

The stability of the system is defined at the frequency where the open loop gain of the amplifier intercepts the closed loop gain response. At this point, the theoretical phase shift of the system should be greater than -180 degrees. In practice, the system phase response should be larger than -135 degrees. This technique is illustrated in Figures 11 through 14. The cases presented in Figure 11 and Figure 12 represent stable systems. The cases presented in Figure 13 and 14 represent unstable systems.

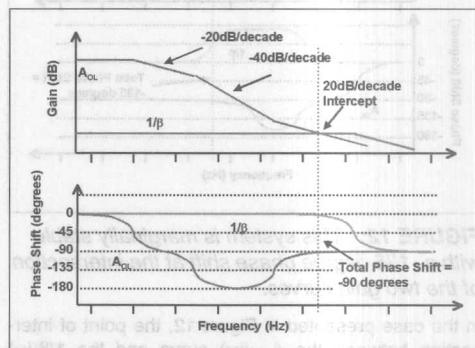


FIGURE 11: This closed loop system is stable with a phase shift of -90 degrees at the intercept of the A_{OL} and $1/\beta$ curves.

In Figure 11, the open loop gain of the amplifier ($A_{OL}(j\omega)$) starts with a zero dB change in frequency and quickly changes to a -20dB/decade slope. At the frequency where the first pole occurs, the phase shift is -45 degrees. At the frequency one decade above the first pole, the phase shift is approximately -90 degrees. As the gain slope progresses with frequency, a second pole is introduced causing the open loop gain response to change -40dB/decade. Once again, this is accompanied with a phase change. The third incident that occurs in this response is where a zero is introduced and the open loop gain response returns back to a -20dB/decade slope.

$1/\beta$ change with frequency. $1/\beta$ remains flat with increased frequency until the very end of the curve where a pole occurs and the curve starts to attenuate $-20\text{dB}/\text{decade}$.

The point of interest in this graph is where the $A_{OL}(j\omega)$ curve intersects the $1/\beta(j\omega)$ curve. The rate of closure of $20\text{dB}/\text{decade}$ between the two curves suggests the phase margin of the system and in turn predicts the stability. In this situation, the amplifier is contributing a -90 degree phase shift and the feedback factor is contributing a zero degree phase shift. The phase shift and consequently the stability of the system is determined at this intersection point. The system phase shift is calculated by subtracting the $1/\beta(j\omega)$ phase shift from the $A_{OL}(j\omega)$ phase shift. In this case, the system phase shift is -90 degrees. Theoretically, a system is stable if the phase shift is between zero and -180 degrees.

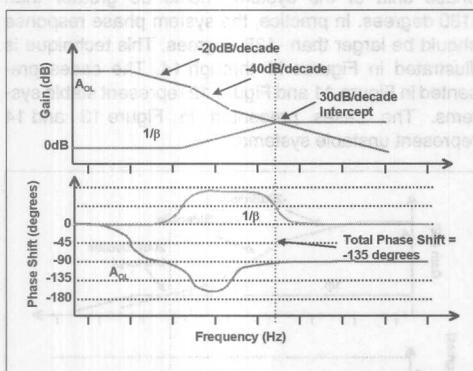


FIGURE 12: This system is marginally stable with a -135 degree phase shift at the intersection of the two gain curves.

In the case presented in Figure 12, the point of intersection between the $A_{OL}(j\omega)$ curve and the $1/\beta(j\omega)$ curve suggests a marginally stable system. At that point, the $A_{OL}(j\omega)$ curve is changing $-20\text{dB}/\text{decade}$. The $1/\beta(j\omega)$ curve is changing from a $+20\text{dB}/\text{decade}$ to a $0\text{dB}/\text{decade}$ slope. The phase shift of the $A_{OL}(j\omega)$ curve is -90 degrees. The phase shift of the $1/\beta(j\omega)$ curve is $+45$ degrees. The system phase shift is equal to -135 degrees.

Although this system appears to be stable, i.e. the phase shift is between zero and -180 degrees, circuit implementation will not be as clean as calculations or simulations would imply. Parasitic capacitance and inductance on the board can contribute additional phase errors. Consequently, this system is deemed "marginally stable" with this magnitude of phase shift. This closed loop circuit will be shown to have significant overshoot and ringing in the Time Response discussion.

$-20\text{dB}/\text{decade}$. The $1/\beta(j\omega)$ is changing at a rate of $+20\text{dB}/\text{decade}$. The rate of closure of these two curves is $40\text{dB}/\text{decade}$ and the system phase shift is -168 degrees. The stability of this system is very questionable.

In Figure 14, $A_{OL}(j\omega)$ is changing at a rate of $-40\text{dB}/\text{decade}$. The $1/\beta(j\omega)$ is changing at a rate of $0\text{dB}/\text{decade}$. The rate of closure of these two curves is $40\text{dB}/\text{decade}$ indicating a phase shift of -170 degrees. The stability of this system is also questionable.

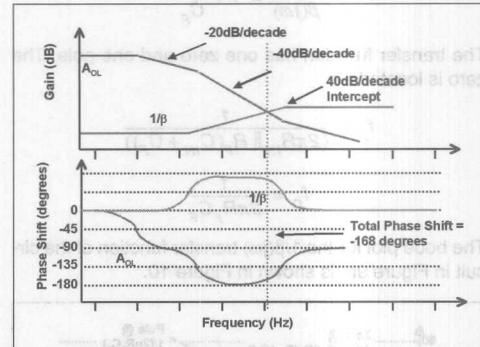


FIGURE 13: In a practical circuit implementation, given layout parasitics, this system is unstable.

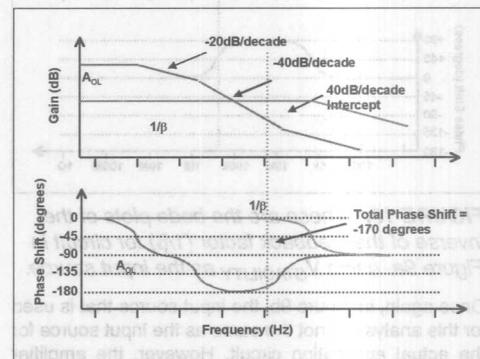


FIGURE 14: In a practical circuit implementation, given layout parasitics, this system is also unstable.

Load Capacitance (C_L) - Output Impedance (Z_0)

A primary application for amplifiers is to separate or isolate a signal source from a load. Sometimes the loads that need to be isolated are fundamentally resistive. In other cases, the load is pure capacitive. The third scenario is where the amplifier has an R/C load. An example of an amplifier circuit that has a resistive

and capacitive load is shown in Figure 15. The critical specification that effects the stability of this system is the open loop output impedance of the amplifier, R_O and the open loop phase response.

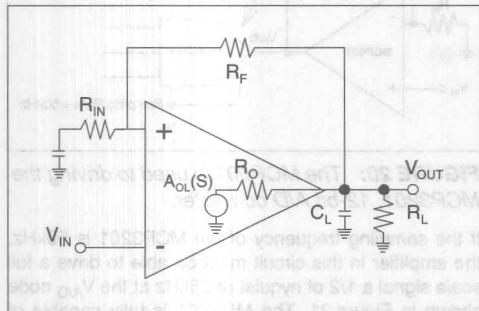


FIGURE 15: Any load on an amplifier may effect the performance of the closed loop circuit. In the case of capacitive loads, the frequency stability can also be compromised.

When the output impedance of the amplifier is considered in this closed loop system, the effective open-loop gain of the amplifier is changed to:

$$A_{OL}(j\omega) = A_{OL}(j\omega) \left(\frac{R_P}{R_P + R_O} \right) \left(\frac{1}{1 + R_X C_L(j\omega)} \right)$$

Where:

$$R_P = R_F \parallel R_L \text{ and}$$

$$R_X = R_O \parallel R_F \parallel R_L$$

Application Challenge - Even though an amplifier is unity gain stable, a capacitive load may make the system unstable.

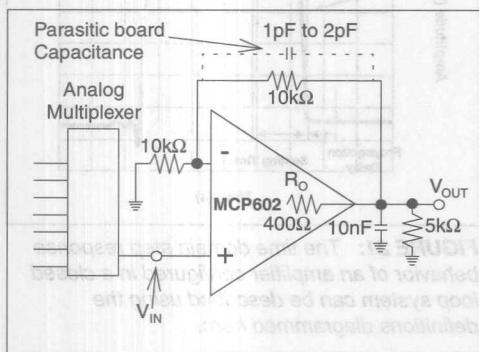


FIGURE 16: This amplifier configuration is potentially unstable because of the low closed loop gain and high capacitive load.

An example where a unity gain stable amplifier can be made unstable is shown in Figure 16. The open loop output resistance is 400Ω . Using the formulas of R_P and R_X above:

$$R_P = 10k\Omega \parallel 10k\Omega = 2.5k\Omega$$

$$R_X = 400\Omega \parallel 10k\Omega \parallel 5k\Omega = 357\Omega$$

The bode plot for the modified open loop (A_{OL}') gain and phase curves are shown in Figure 17, along with the actual A_{OL} plots of the amplifier.

The feedback transfer function is:

$$1/\beta = 1 + 10k\Omega / 10k\Omega$$

The bode plot of $1/\beta$ is also shown in Figure 17.

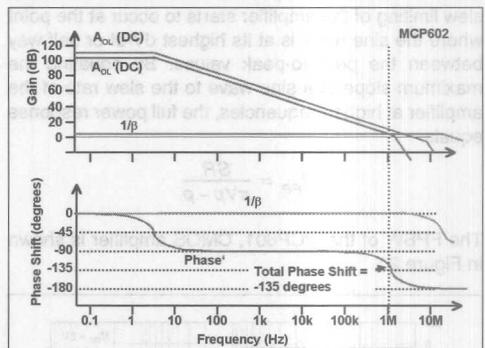


FIGURE 17: Capacitive loads on amplifiers degrade the system stability. This bode plot demonstrates the effects of the load illustrated in Figure 16.

From this simple calculation, the circuit is shown to be very marginal in terms of instability. Any parasitic board capacitance will simply aggravate the condition.

Capacitive loading can be corrected by implementing the circuit shown in Figure 18.

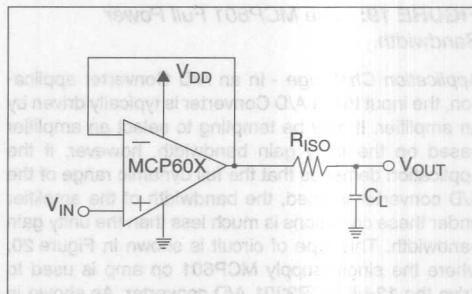


FIGURE 18: In most cases, the addition of a resistor, R_{ISC} , between the output of the amplifier and the capacitive load will eliminate any unwanted oscillation.

Full Power Bandwidth

Specification Discussion - The Full Power Bandwidth (FPBW) of an amplifier is the maximum frequency that the output of an amplifier can swing over the full dynamic range without significant distortion. At lower frequencies, the FPBW is limited by the output swing of the amplifier. At higher frequencies, the response is limited by the slew rate of the amplifier. The definition of slew rate is given under the Time Domain specifications later on in this application note. Distortion due to slew limiting of the amplifier starts to occur at the point where the sine wave is at its highest dV/dt or half way between the peak-to-peak values. By equating the maximum slope of a sine wave to the slew rate of the amplifier at higher frequencies, the full power response equals:

$$f_{FP} = \frac{SR}{\pi V_{P-P}}$$

The FPBW of the MCP601, CMOS amplifier is shown in Figure 20.

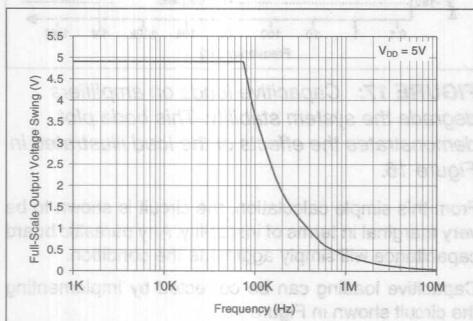


FIGURE 19: The MCP601 Full Power Bandwidth.

Application Challenge - In an A/D Converter application, the input to the A/D Converter is typically driven by an amplifier. It may be tempting to select an amplifier based on the unity gain bandwidth, however, if the application demands that the full dynamic range of the A/D converter is used, the bandwidth of the amplifier under these conditions is much less than the unity gain bandwidth. This type of circuit is shown in Figure 20, where the single supply MCP601 op amp is used to drive the 12-bit MCP3201 A/D converter. As shown in Figure 19, the FPBW of the MCP601 is 80kHz. In contrast, the unity gain bandwidth (which is a small signal specification) is typically 2.8MHz.

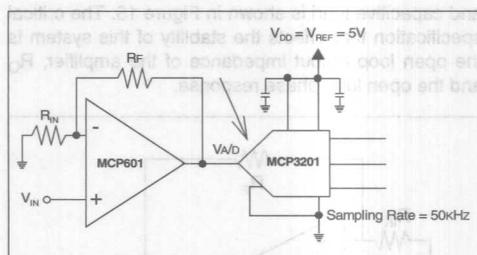


FIGURE 20: The MCP601 is used to driving the MCP3201, 12-bit A/D converter.

If the sampling frequency of the MCP3201 is 50kHz, the amplifier in this circuit must be able to drive a full scale signal 1/2 of nyquist or 25kHz at the $V_{A/D}$ node shown in Figure 21. The MCP601 is fully capable of meeting this performance requirement with a FPBW that exceeds 25kHz by more than three times.

TIME DOMAIN SPECIFICATIONS

The time domain responses of amplifier circuits provide a real world result of the previous frequency discussions. The graphical definition of time domain specifications is shown in Figure 21. The waveform in this figure depicts the response of the output of the amplifier with regards to a step response at the input of the circuit. This figure will be referred to throughout the following discussion.

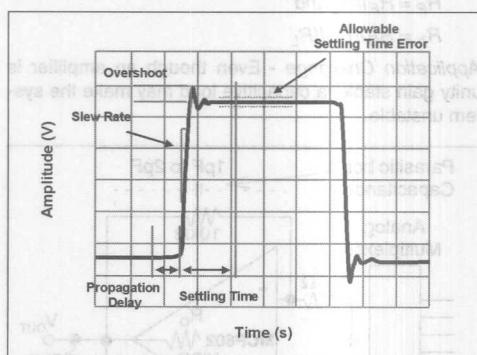


FIGURE 21: The time domain step response behavior of an amplifier configured in a closed loop system can be described using the definitions diagrammed here.

Slew Rate (SR)

Specification Discussion - The Slew Rate (SR) specification for an amplifier quantifies the speed at which the output terminal can execute a full scale output voltage swing that is driven at the input of the amplifier. This specification is controlled internally by the amplifier and dependent on the amount of tail current available to charge and discharge internal capacitors. The units of

this specification are in volts per second. Slew rate is measured from 10% to 90% at the output of the amplifier through the full scale voltage swing. The most challenging amplifier circuit for this type of specification is the buffer or follower configuration, as shown in Figure 22. In this configuration, the input terminals are pulled out of their nonlinear region by virtue of the fact that the inverting input terminal is connected directly to the slow moving output terminal.

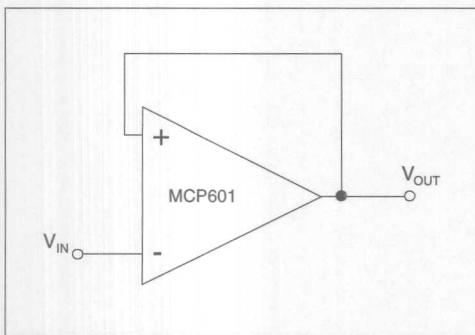


FIGURE 22: The most challenging test condition for slew rate is the buffer configuration that is shown here.

Settling Time (t_s) and Overshoot

At the top or bottom of the full-scale slew transition, a degree of ringing occurs. This ringing is directly related to the phase shift of the closed loop system and described in terms of overshoot and the amount of time before the signal to settle within a specified error band. The % overshoot is defined at the highest peak in this portion of the waveform, which occurs at the beginning of the ringing of the system. The magnitude of the system overshoot and the amount of time required for the system to settle is directly related to the frequency domain phase shift of the system.

The Settling Time (t_s) of an amplifier circuit is defined as the amount of time that is required for the output of the amplifier to slew and then settle within a defined error band. This time starts at the point where the output first responds to the input excitation until the last occurrence of the output signal being outside the error band, as illustrated in Figure 21.

Application Challenge - A t_s error is most noticeable in applications where the common mode input of the amplifier has a full scale step response applied. In this situation, the amplifier goes into a full slew rate condition (SR) and then settles to its final value (t_s). For this example, the circuit shown in Figure 16 is integrated into the circuit shown in Figure 23. Even though the system signals are slow moving, the multiplexer in this circuit presents a step response to the amplifier. The step response of this amplifier system is shown in Figure 24.

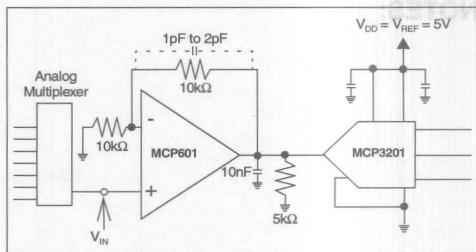


FIGURE 23: The driver configuration shown in Figure 16 is used in this multiplexing circuit. This driver circuit does not make the system unstable, however, the A/D converter conversion should be delayed while the op amp circuit is settling to final value.

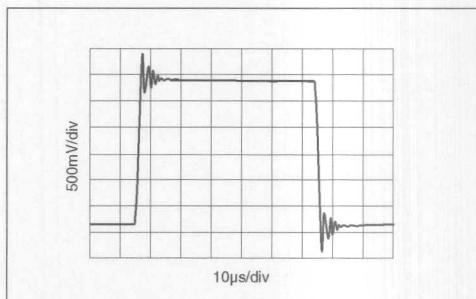


FIGURE 24: The phase shift of an amplifier in a closed loop system is reflected in the step response in terms of overshoot (15%) and ringing.

REFERENCES

- Wait, Huelsman, Korn, "Introduction to Operational Amplifier Theory and Applications", McGraw Hill, 1975.
- Baker, Bonnie C, "Operational Amplifier Topologies and DC Specifications", AN-722, Microchip Technology, Inc.
- Ott, Henry W., "Noise Reduction Techniques in Electronic Systems", John Wiley & Sons, 1988.
- Frederiksen, Thomas M., "Intuitive Operational Amplifiers", McGraw Hill, 1988.

NOTES:

Using the MCP2510 CAN Developer's Kit

Author: Pat Richards
Microchip Technology Inc.

INTRODUCTION

The MCP2510 eases software development and shortens the learning curve for the MCP2510 by providing three PC software templates with different functions that CAN node designers can use as needed. By using one of the templates, the user can read, display, and modify all of the registers in the MCP2510 down to the bit level. Another template can demonstrate basic input/output functionality by allowing messages to be transmitted and received on the CAN bus. Finally, the third template is a demonstration template that, by using a preprogrammed PICmicro®, implements a two node CAN bus that gets the user "on the bus" in the shortest amount of time and effort.

This application note serves as a three-part tutorial for the MCP2510 and discusses the three software templates in detail as well as the important menu items.

Figure 1 is a block diagram of the MCP2510 board. The board is a two node CAN system that can be used together to implement a simple bus or can be con-

nected to an external bus using the off-board connector. For the purposes of this application note, the left node will be referred to as the **PC node** and the right node will be referred to as the **PICmicro® node**.

Both nodes are identical except for the controller interface to the MCP2510 is a PC for one node and a PIC microcontroller for the other node. By providing a two node CAN network on a single board, a simple CAN network can be implemented with one PCB.

PC Node

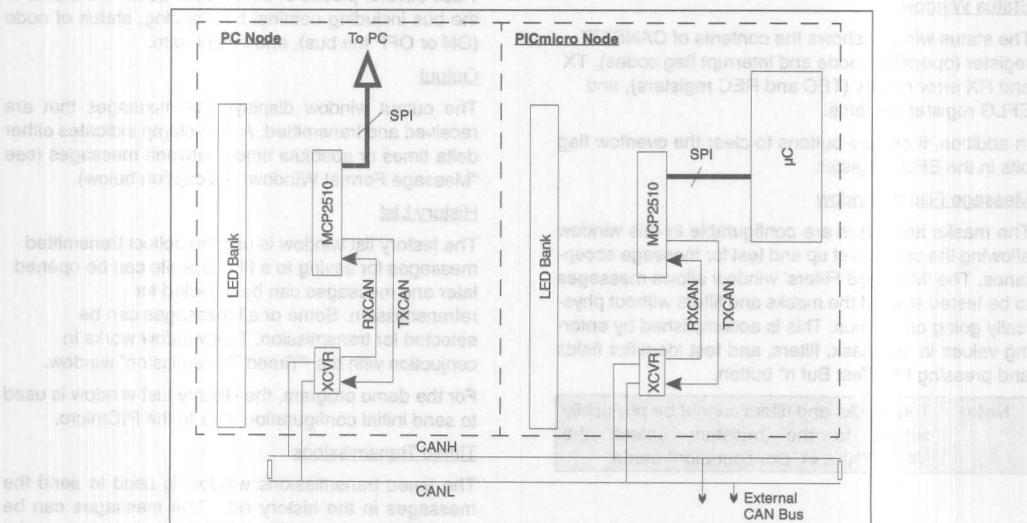
The left node (as oriented in Figure 1) is referred to as the PC node. The PC performs the microcontroller functions and interfaces to the MCP2510 via Serial Peripheral Interface (SPI™) through the parallel port.

The software templates control this node.

PICMicro Node

The right node is referred to as the PICmicro node and represents a typical CAN node. This node is used by the end user as needed. When running the "MCP2510 Evaluation" template, demonstration firmware is used to complete a CAN system.

FIGURE 1: BLOCK DIAGRAM OF MCP2510 DEVELOPMENT BOARD



SPI is a trademark of Motorola Inc.

OVERVIEW OF THE SYSTEM SOFTWARE

There are three software templates that are used to control one or both of the nodes on the board. This tutorial will discuss all three of the templates. Refer to the User's Guide for images of the windows if needed.

Note: The templates can be saved as projects which will save the layout and default properties by selecting File > Save. Saving the project does not save the register contents of the MCP2510. Saving the register contents can be done in the Register Template as discussed at the end of the Register Template tutorial (Part 3).

The following is a brief discussion of the three templates:

Register Template

The "Register" template is a low level template that allows bit-level control of the MCP2510 registers. This template may be used to become familiar with the MCP2510 by experimenting with mask and filter settings, bit timings, configuration registers, and other functions associated with configuring the MCP2510.

There are several windows associated with the Register Template:

MCP2510 Evaluation Board Window

This window displays the connection status, parallel port address, and MCP2510 mode of operation. The MCP2510 can be reset in this window and MCP2510 register contents be saved and loaded. The register contents are saved as a HEX file.

Status Window

The status window shows the contents of CANSTAT register (operation mode and interrupt flag codes), TX and RX error counts (TEC and REC registers), and EFLG register contents.

In addition, there are buttons to clear the overflow flag bits in the EFLG register.

Message Filters Window

The masks and filters are configurable in this window allowing the user to set up and test for message acceptance. The 'Message Filters' window allows messages to be tested against the masks and filters without physically going on the bus. This is accomplished by entering values in the mask, filters, and test identifier fields and pressing the "Test Buf n" button.

Note: The masks and filters cannot be physically written to the registers unless the MCP2510 is in "Configuration" mode.

Physical Layer Window

The three CNF registers used for all CAN bit timings are configured in this window.

Note: The CNF registers can only be modified while the MCP2510 is in "Configuration" mode.

Configuration Window

TXRTSCTRL, BFPCTRL, CANINTF, CANINTE, and CANCTRL are all modified from this window. These are the control and flag registers.

If a message is received while in the Register Template, the receive buffer flags in CANINTF (RX0IF and RX1IF) must be cleared manually to receive additional messages.

Transmit Window

The transmit window controls the buffer contents for the transmit registers including TXBnCTRL, the identifier registers, and the data registers.

Receive Window

This window contains all of the buffer contents for the receive buffers including RXBnCTRL, the identifier registers, and the data registers.

Basic Template

This template provides high-level control of one node (PC node) and is identical to the "MCP2510 Evaluation" template minus the window for the second node (see the tutorial part 1). This template can be thought of as a simple bus monitor with transmit capabilities.

Bus Status

This window, labeled 'MCP2510 CAN Controller', provides several pieces of information about the status of the bus including nominal bus loading, status of node (ON or OFF the bus), and bus bit rate.

Output

The output window displays the messages that are received and transmitted. A time stamp indicates either delta times or absolute times between messages (see "Message Format Window" description below).

History List

The history list window is used to collect transmitted messages for saving to a file. This file can be opened later and messages can be selected for retransmission. Some or all messages can be selected for transmission. This window works in conjunction with the "Timed Transmission" window.

For the demo program, the History List window is used to send initial configuration data to the PICmicro.

Timed Transmissions

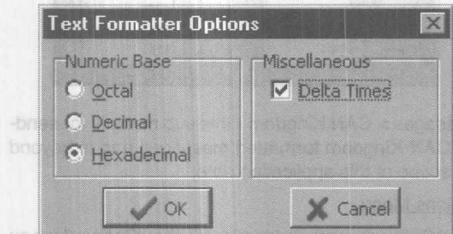
The timed transmissions window is used to send the messages in the history list. The messages can be sent either one time (one shot) or repeated at regular intervals (cyclic).

Message Format Window

This window determines the format of the displayed data in the output window. The default is 'Standard Text Format' which displays the message data as normal data.

To change the property of the "Standard Text Format", highlight "Standard Text Format" and press the "Properties" button. The numeric base and the time display can be changed as indicated in Figure 2.

FIGURE 2: TEXT FORMAT FOR OUTPUT WINDOW

**MCP2510 Evaluation Template**

This template contains a simple demonstration program. Only high-level control is possible on the PC node. Indirect control of the PICmicro node is made possible via the CAN bus (i.e., the PICmicro node is configured by the PC node via the CAN bus). The demo program incorporates the CAN Kingdom Higher Layer Protocol (HLP). A specific setup procedure (covered later) is required to enable the two nodes to communicate with each other.

The windows associated with the Evaluation Template are the same as the Basic Template with the following addition:

MCP2510 Eval Board Controls Window

This window, combined with the "Output" window, is the main window of interest for the demo. This window displays the graphical representation of the messages on the CAN bus.

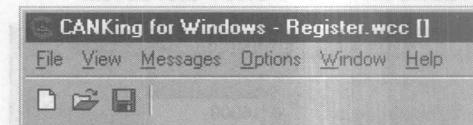
UNDERSTANDING THE MENUS

There are several menu items that are discussed in this application note. The remaining menu items that are not discussed are the self explanatory menus such as "File > Save" or "Help". The menus change slightly between the three software templates as discussed.

Register Template Menus

The menu items for the Register Template is shown in Figure 3.

FIGURE 3: MENU BAR



The three menus that are discussed are:

View Menu

This menu makes visible/invisible, the windows loaded in the Register Template.

Messages Menu

This menu loads windows associated with the Register Template.

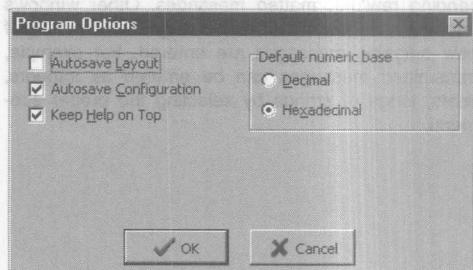
Note: Selecting a window while holding down the shift key will load a copy of the window so that more than one window of the same type can be visible. This is useful for opening multiple "Receive" and "Transmit" windows for viewing multiple buffers simultaneously.

Options Menu

The "Options" menu contains two sub-menu items that, as the name implies, sets options for the software.

Options > Global... - as shown in Figure 4 has autosave options and numeric base settings. The numeric base settings select the default numeric base as displayed/entered in the windows. Selecting hexadecimal automatically places a '\$' in front of the entered number indicating the number entered is HEX.

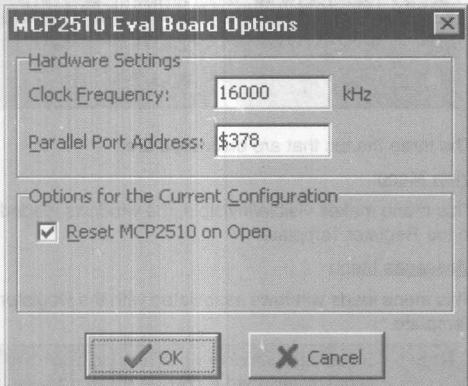
FIGURE 4: OPTIONS > GLOBAL...



Options > MCP2510... - as shown in Figure 5 sets the clock frequency, parallel port address, and whether the MCP2510 is reset on software powerup.

The clock frequency and parallel port address must be set properly for the software to function properly. The MCP2510 clock frequency is used in calculating the CAN bus rate and is therefore required to be set. The parallel port address much match the BIOS address setting. Typical values for the parallel port are 0x3BC, 0x378, and 0x278. Address 0x378 is the most common default for LPT1.

FIGURE 5: OPTIONS > MCP2510...



Basic and Demonstration Template Menus

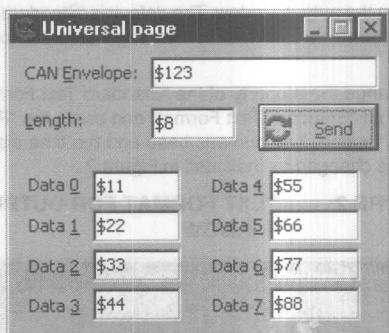
The Basic and Demonstration Template menus are identical as shown in Figure 3. The sub-menus are same for the Basic and Demonstration Templates, however, they differ from the Register Template sub-menus as follows:

Messages Menu

The "Messages" contains two sub menus that select transmit message formats.

Messages > Universal > Universal - (shown in Figure 6) loads a window used for transmitting messages onto the CAN bus. The identifier, data length, and data are entered and sent with this window. The "Universal" window can be thought of as a window for sending raw/unformatted messages. Other windows under "Messages > Universal" change the formatting of how outgoing messages are entered. For example, transmitted messages can be entered as integers, floats, longs or strings by selecting the proper submenu.

FIGURE 6: MESSAGES > UNIVERSAL >UNIVERSAL



Messages > CAN Kingdom - this sub menu is for sending CAN Kingdom formatted messages and is beyond the scope of this application note.

Options Menu

The "Options" menu is identical to the "Options" menu in the "Register Template" except for the addition of a "King's Pages" sub-menu which is beyond the scope of this application note.

Options > King's Pages - this sub menu is for sending CAN Kingdom formatted messages and is beyond the scope of this application note.

Options > MCP2510 - this sub menu is for sending MCP2510 formatted messages and is beyond the scope of this application note.

Options > Parallel Port - this sub menu is for sending Parallel Port formatted messages and is beyond the scope of this application note.

TUTORIAL PART 1: SETTING UP AND RUNNING THE DEMO

The MCP2510 CAN Development Kit demo utilizes both of its CAN nodes to demonstrate basic CAN communications. Each node is set up to transmit and receive messages.

Initial Setup

A few initial setup procedures should be performed to insure proper operation before continuing.

Verify/Set the Parallel Port Address

The parallel port address must match the operating system BIOS setting. To set/verify, select "Options > MCP2510...".

Set Oscillator Frequency in Software

The oscillator frequency must be set to match the board oscillator so the CAN bit rate will be reflected accurately. To set, select "Options > MCP2510..." and set to the proper oscillator frequency (16000 kHz if using the 16 MHz oscillator supplied with the kit).

Save Configuration

Saving the configuration as a project insures that the new settings are saved. To save, select "File > Save" and name the project something descriptive.

Reopen Saved Project

This refreshes the settings to the saved values. To reopen, select "File > Reopen" and choose the file that was just saved.

Running the Demo

Open a new "MCP2510 Evaluation Board" project template (File > New).

Refer to Figure 7 for the configuration procedure. After stepping through the configuration procedure, the two nodes should be communicating with each other. A receive message (received by the PC node) should be displayed in the "Output" window at every timer interval as set in the "MCP2510 Eval Board Controls" window (\$100 or 256 ms in the case of the flow diagram). The data contains the potentiometer value and the S4, S5, S6 button status as read by the PICmicro node.

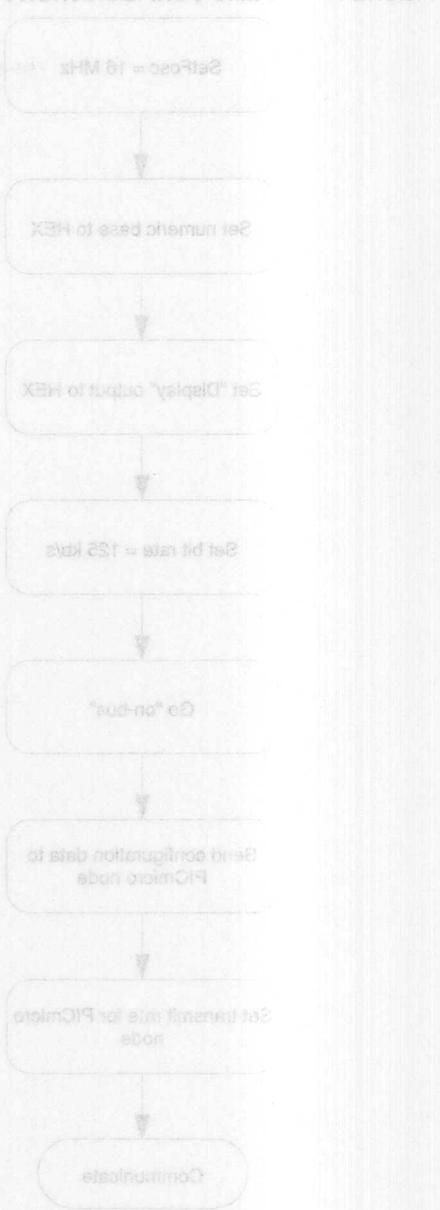
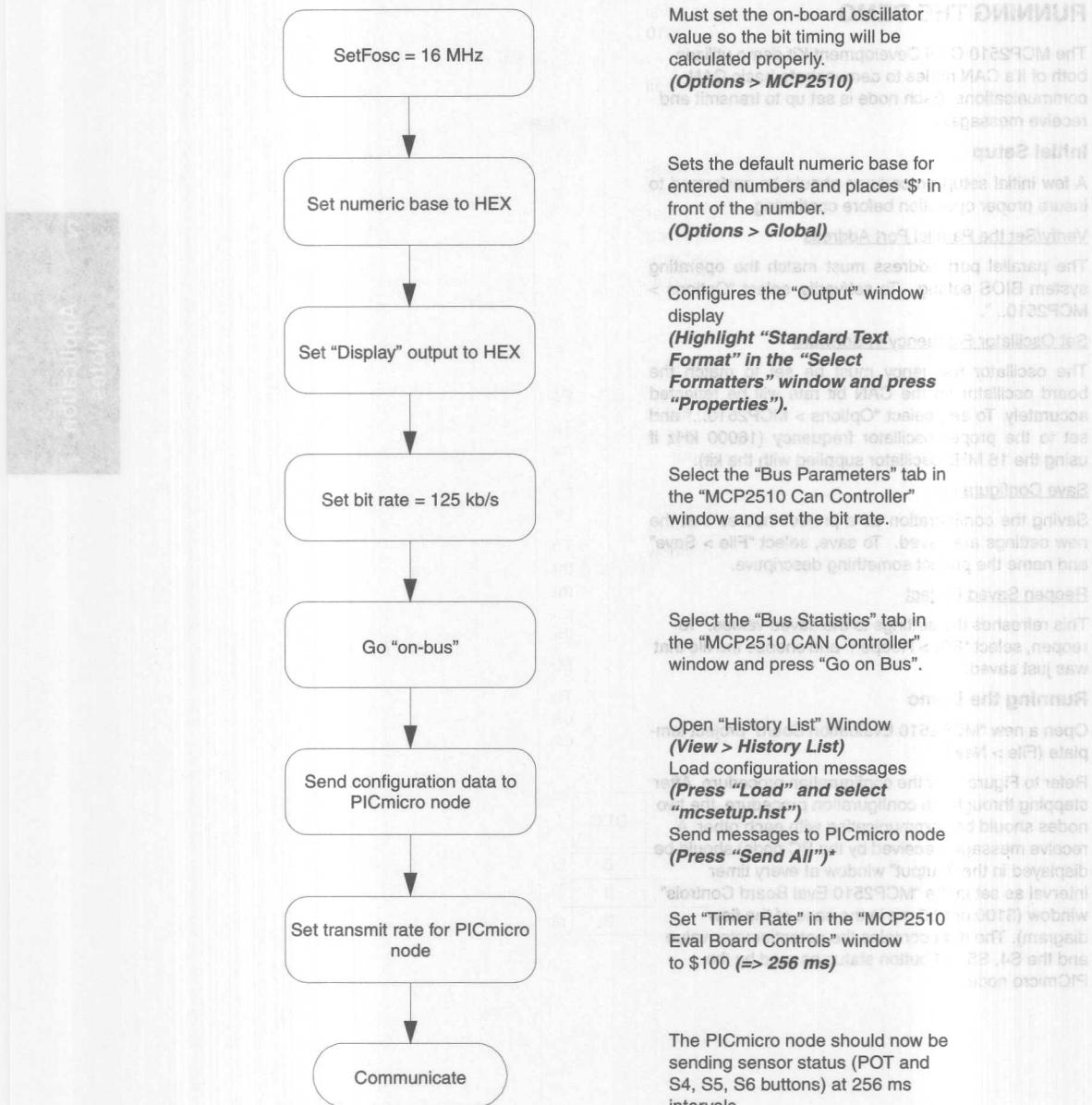


FIGURE 7: DEMO CONFIGURATION PROCEDURE



* The "Output" window should display some transmitted messages and two received messages. If both receive messages are not displayed, reset the nodes by pressing $MCLR$ on the PICmicro node and "Reset Board" button on the PC node and goto the "Go on-bus" step.

Two transmit messages can be displayed in the "Output" window by checking/unchecking the LED boxes or by pressing the "Set" button to set the transmit interval time. These controls are contained in the "MCP2510 Eval Board Controls" window.

The function of each node is explained in more detail below.

PC Node

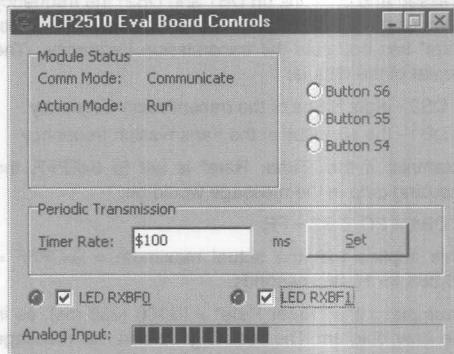
The PC node is set up to receive and display all messages sent by the PICmicro node (in the "Output" window). The received messages contain potentiometer data and button status (S4, S5, S6) and are displayed graphically in the "MCP2510 Eval Board Controls" window. The "raw" message is displayed in the "Output" window.

The PC node can transmit two message types:

1. The timer rate that the PICmicro node uses to schedule its message transmissions.
2. The digital output of the RXnBF pins, which are indicated on the associated LEDs. These pins on the MCP2510 are configured as general purpose outputs for the purpose of the demo.

The messages are transmitted by either pressing the "Set" button or checking/unchecking the LED checkboxes in the "MCP2510 Eval Board Controls" window (Figure 8).

FIGURE 8: MCP2510 EVAL BOARD CONTROLS



PICMicro Node

The PICmicro node is set up to respond to messages received by PC node. This response will either involve changing its own transmit interval or driving the RXnBF pins which are reflected on corresponding LEDs.

The PICmicro node will transmit a single message at the predefined transmit interval (\$100 ms, or 256 ms for the tutorial) which contains the ADC value from the potentiometer in byte 1. This message also contains the button status for S4, S5, and S6 in byte 0.

Message Formats

There are three different message identifiers that are used in the system implementation. Each identifier indicates a specific function as explained in Table 1.

TABLE 1: MESSAGE FORMATS

Message Definition	ID	DLC	Data	Message Direction
Scheduled Transmission Frequency	0066	8	00 nn nn 00 00 00 00 00	PC to PICmicro
LED Control (RXBFn)	0067	8	0n 00 00 00 00 00 00 00	PC to PICmicro
Potentiometer and Buttons	0069	2	nn nn	PICmicro to PC

n - represents changing data

Scheduled Transmission Frequency (message 066)

This message is sent from the PC node when "Set" is pressed and contains (in DB1 and DB2) the frequency in which the PICmicro node is to send data. The "Timer Rate" text box sets the transmission frequency. The format of the data is:

- DB2 - eight MSbs of the transmission frequency
- DB1 - the six LSbs of the transmission frequency

Example: if the "Timer Rate" is set to 0xFFFF, the resulting data in the message would be:

- DB1 = FC, DB2 = FF.

This implies that the actual maximum = 0x3FFF = 16,383 => 16.383 seconds.

Example: if the "Timer Rate" = 0x100 (256 ms), as in the flow diagram The resulting data in the message would be DB1 = 00, DB2 = 40.

LED Control (message 067)

This message is sent from the PC node and is used to control the RXFBn LEDs on the PICmicro node.

The format of the data is:

- LED RXFB0 - DB0, bit0; (0 = OFF, 1 = ON)
- LED RXFB1 - DB0, bit1, (0 = OFF, 1 = ON)

Potentiometer and Buttons (message 069)

This message is scheduled and is sent from the PICmicro node. The message is sent at regular intervals as determined by message '066' and contains the digital equivalent of the potentiometer and the button status for the three buttons associated with the PICmicro node. For each button, there is one bit showing the current status and one sticky bit that is set if the button has been pressed since the last poll.

The format of the data:

- DB0 - button status (six MSbs => three status bits and three sticky bits).
- DB1 - analog to digital converted value.

DB0	DB1
00 00 00 00 00 00	00 00 00 00 00 00
00 00 00 00 00 00	00 00 00 00 00 00
00 00 00 00 00 00	00 00 00 00 00 00

TUTORIAL PART 2: THE BASIC TEMPLATE

The "Basic" template is identical to the "MCP2510 Evaluation" template in every aspect except that the "MCP2510 Eval Board Controls" is not available in the "Basic" template. Therefore, this tutorial will focus on the different settings/configurations available.

Note: If the Demo Tutorial (Part 1) was set up prior to this one, the PICmicro node is probably still sending messages at regular intervals. Setting the CAN bit rate to 125 kb/s and going on the bus will display these messages in the "output" window.

Changing the Output Window

The "Output" window is the main window of activity in the "Basic" template and can be modified to different display properties.

Scrolling and Fixed Messages

The messages can be displayed in either scrolling or fixed format simply by clicking the right mouse button while the cursor is over the "Output" window and selecting the proper display properties (Figure 9).

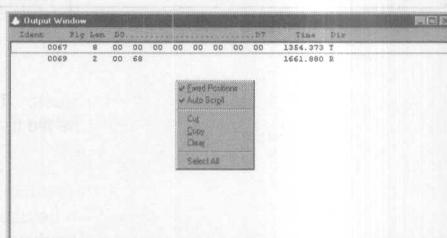
Scrolling Messages - deselect "Fixed Positions". Selecting "Auto Scroll" automatically scrolls the window to the newest message.

Non-scrolling Messages - Select "Fixed Positions". This selection anchors messages with matching identifiers into one position; only the data and time are updated.

Changing the Time Base

The "Time" can be displayed as either free-running time between messages or as delta time by highlighting "Standard Text Format" in the "Select Formatters" window and pressing the "Properties" button (see Figure 2).

FIGURE 9: CHANGING THE OUTPUT WINDOW DISPLAY PROPERTIES



TUTORIAL PART 3: THE REGISTER TEMPLATE

As described earlier, the "Register" template is a low level template that allows bit-level control of the MCP2510 registers. This template may be used to become familiarized with the MCP2510 by experimenting with mask and filter settings, bit timings, configuration registers, and other functions associated with configuring the MCP2510.

This tutorial will discuss the associated windows and provide examples for some. In addition, a Loopback mode demonstration will be given to demonstrate the basic steps required for successful CAN communications with the MCP2510.

Note: Unlike the "Basic" and "Evaluation" templates which modify specific registers automatically, all registers of the MCP2510 while in the "Register" template are controlled manually (e.g. transmit requests are set manually and the receive flags must be cleared manually for each message sent and received). In essence, the user is the microcontroller while in the "Register" template.

Initial Setup

A few initial setup procedures should be performed to insure proper operation before continuing.

Verify/Set the Parallel Port Address

The parallel port address must match the operating system BIOS setting. To set/verify, select "Options > MCP2510..."

Set Oscillator Frequency in Software

The oscillator frequency must be set to match the board oscillator so the CAN bit rate will be reflected accurately. To set, select "Options > MCP2510..." and set to the proper oscillator frequency (16000 kHz if using the 16 MHz oscillator supplied with the kit).

Save Configuration

Saving the configuration as a project insures that the new settings are saved. To save, select "File > Save" and name the project something descriptive.

Reopen Saved Project

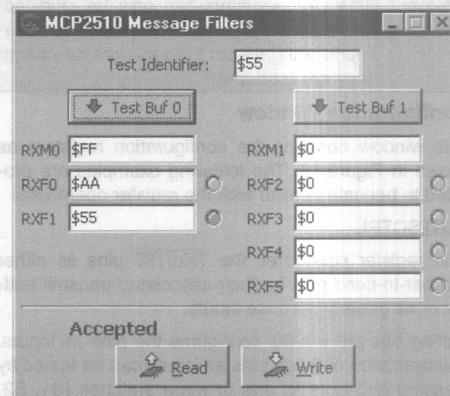
This refreshes the settings to the saved values. To reopen, select "File > Reopen" and choose the file that was just saved.

Mask and Filter Settings

The "MCP2510 Message Filters" window can be used to experiment with various mask and filter configurations and how these configurations apply to incoming message IDs. For example, setting "RXM0", "RXF0", "RXF1", and "Test Identifier" as shown in Figure 10 would reveal that the Test Identifier would match RXF1 and be accepted into buffer 0. Pressing the "Test Buf n" buttons is a software test of the masks and filters.

Refer to the MCP2510 data sheet for more details on mask and filter operations.

FIGURE 10: MCP2510 MESSAGE FILTERS



When finished setting the mask and filters, the MCP2510 can be written by pressing the "Write" button while in "Configuration" mode.

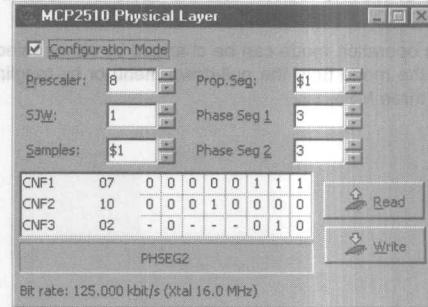
Important Considerations

- The mask/filter and CNF registers are not written unless the "Write" button is pressed while the MCP2510 is in Configuration mode.
- An 'x' at the end of a value indicates "extended" identifier (29 bits). If a filter value has an 'x', the incoming identifier will be accepted only if it also is extended (contains an 'x').
- The '\$' represents HEX
- Standard identifiers are 11-bits with a range of \$0 - \$7FF.
- Extended identifiers are 29-bits with a range of \$0 - \$FFFFFFE.

Bit Timing

The "MCP2510 Physical Layer" window contains the three CNF registers which are used to set the CAN bit rate. The resulting bit rate is displayed at the bottom of the window for a given Fosc and CNF register values.

FIGURE 11: PHYSICAL LAYER WINDOW



Example, Figure 11 shows one possible configuration of the bit timing registers for a CAN bit rate of 125 kb/s using a 16 MHz oscillator.

Note: The CNF registers can only be modified while in "Configuration" mode. Any other mode will gray out the registers indicating they are read-only.

Configuration Window

This window contains the configuration registers as shown in Figure 12. The following examples are provided to become familiar with the register operations:

TXRTSCTRL

This register configures the TXnRTS pins as either request-to-send pins for their associated transmit buffers or as general purpose inputs.

Setting bits 2:0 = b'00' configures the pins as inputs. The operation of these pins as inputs can be tested by pressing and holding one or more switches (S1, S2, S3) and pressing "Read". Bits 5:0 will reflect the status of the buttons.

BFPCTRL

This register configures the two RXnBF pins as receive buffer full pins, as general purpose outputs, or tristate (off).

Setting bits 3:2 = b'11' and bits 1:0 = b'00' configures the pins as outputs. The associated RXBF_n LEDs can be toggled on/off by changing the values of bits 5:4 and clicking "Write".

CANINTE

CANINTE is the interrupt enable register that routes its associated flag bit in CANINTF to the INT pin. The flag bits in CANINTF can be set/cleared regardless of CANINTE.

To demonstrate the interrupts, simply set a bit in CANINTE and toggle its associated bit in CANINTF. The INT pin LED should toggle with the flag bit.

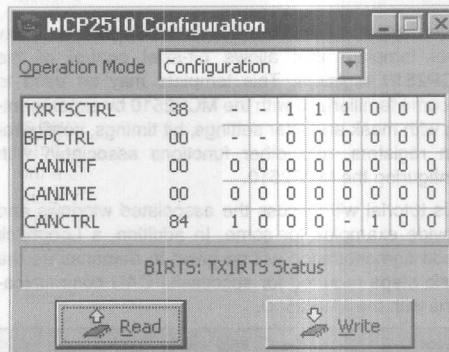
Example: Set CANINTE.RX0IE = 1 (bit 0) and then toggle CANINTF.RX0IF (bit 0) and click "Write". The INT LED will toggle with the flag bit.

CANCTRL

The CANCTRL register sets the operation mode and the CLKOUT prescaler.

The operation mode can be changed by either selecting the mode from the pull-down menu or by toggling the three MSbs.

FIGURE 12: CONFIGURATION REGISTERS



Status Window

The "MCP2510 Status" window contains the status of several parameters including the mode of operation, pending interrupts, transmit and receive error counters values, and receive buffer overflow status.

Transmit and Receive Registers

The transmit and receive registers are displayed in two separate windows and contain the identifier, data length code, and data registers. The transmit register window also contains the CTRL (TXBnCTRL) register which contains the Transmit Request (TXREQ) bit (bit 3).

Communicating in LOOPBACK Mode

While "Loopback" mode is primarily a diagnostics mode and is not used in real applications, it can be used to demonstrate communication procedures while in the "Register" template. The following is a step-by-step demonstration of sending and receiving messages while in "Loopback" mode:

1. Set the mask and filters for receive buffer 0 as shown in Figure 10. The demo will send two different messages with IDs of 0x55 and 0xAA. Remember to set "Configuration" mode before attempting to write the masks and filters.
2. Set the CAN bit rate as shown in Figure 11. While it is not necessary to set the bit rate because the MCP2510 is communicating to itself, it is good practice to configure the CNF registers.
3. Write CANINTE to 0x01 to enable the INT pin on messages received into receive buffer 0. This is done only to provide a visible indication on the board of a received message.
4. Make sure CANINTF.RX0IF = 0 (bit 0). Receive buffers can receive messages only when their associated flag is cleared.
5. In the "Transmit Buffer" window, select transmit buffer 0 with the pull-down box. Set "CAN ID" =

\$AA, "Length" (DLC) = 8, and set the data bytes to unique values. This message, when transmitted will match filter 0.

Select Transmit buffer 1 and set the "CAN ID" = \$55, DLC = 8, and the data to unique values. This message will match filter1.

6. The MCP2510 register contents can be saved at this point if desired to be reloaded at a later time. The register contents are saved here before the last step because the MCP2510 is still in "Configuration" mode. To save the register settings, press the "Save Regs" button in the "MCP2510 Evaluation Board" window.
7. In the "Configuration" window, select "Loop-back" mode and click "Write".

The MCP2510 is now set up to transmit and receive its own messages.

1. Select transmit buffer 0 or 1.
2. Toggle CTRL.TXREQ (bit 3) to a '1' and click "Write". The INT LED should be on.
3. Read the "Receive Buffer" window. The transmitted message should be in the receive buffer.
4. Clear CANINTF.RX0IF (bit 0). The INT LED should be off.
5. Goto number 1.

CONCLUSION

This concludes the tutorials demonstrating the usefulness of the MCP2510 CAN Development Kit. With it, one can quickly learn the fundamentals of Controller Area Networks and utilize it for system development.

AN733

NOTES:

SECTION 3 DATASHEETS

Microcontroller Supervisory Circuit with Push-Pull Output - MCP100/101	3-1
Microcontroller Supervisory Circuit with Open Drain Output - MCP120/130	3-9
Microcontroller Supervisory Circuit with Push-Pull Output - MCP809/810	3-15
Stand-Alone CAN Controller with SPI™ Interface - MCP2510.....	3-23
2.7V to 5.5V Single Supply CMOS Op Amps - MCP601/602/603/604.....	3-99
2.5V to 5.5V Micropower CMOS Op Amps - MCP606/607/608/609	3-119
2.7V 10-Bit A/D Converter with SPI™ Serial Interface - MCP3001.....	3-149
2.7V Dual Channel 10-Bit A/D Converter with SPI™ Serial Interface - MCP3002.....	3-167
2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI™ Serial Interface - MCP3004/3008.....	3-185
2.7V 12-Bit A/D Converter with SPI™ Serial Interface - MCP3201	3-205
2.7V Dual Channel 12-Bit A/D Converter with SPI™ Serial Interface - MCP3202.....	3-223
2.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI™ Serial Interface - MCP3204/3208.....	3-241

3

Datasheets

Microcontroller Supervisory Circuit with Push-Pull Output

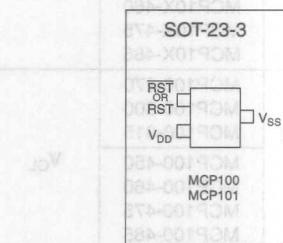
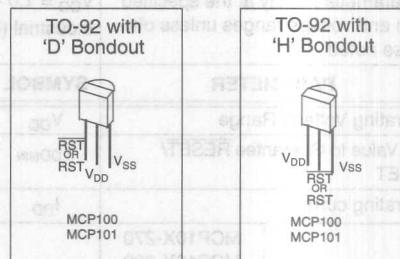
FEATURES

- Holds microcontroller in reset until supply voltage reaches stable operating level
- Resets microcontroller during power loss
- Precision monitoring of 3V, 3.3V, and 5V systems
- 7 voltage trip points available
- Active low RESET pin (MCP100) or active high RESET (MCP101)
- Push-pull output
- Holds RESET/RESET for 350 ms (typical)
- Guaranteed RESET/RESET to $V_{DD} = 1.0V$
- Accuracy of $\pm 125mV$ for 5V systems and $\pm 75mV$ for 3V systems over temperature
- 45 μA typical operating current
- Temperature range:
 - Industrial (I): -40°C to +85°C

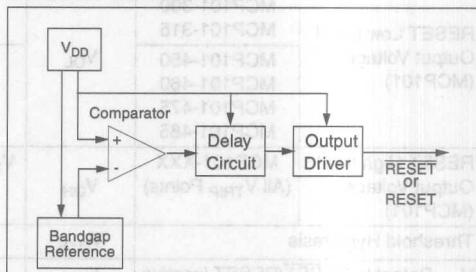
DESCRIPTION

The Microchip Technology Inc. MCP100/101 is a voltage supervisory device designed to keep a microcontroller in reset until the system voltage has reached the proper level and stabilized. It also operates as protection from brown-out conditions when the supply voltage drops below a safe operating level. Both devices are available with a choice of seven different trip voltages and both have push-pull outputs. The MCP100 has a low active RESET pin and the MCP101 has a high active RESET pin. The MCP100/101 will assert the RESET/RESET signal whenever the voltage on the V_{DD} pin is below the trip-point voltage.

PACKAGES



BLOCK DIAGRAM



MCP100/101

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to V_{DD} +1.0V
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	$\geq 2\text{ kV}$

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

DC AND AC CHARACTERISTICS

All parameters apply at the specified temp and voltage ranges unless otherwise noted.		$V_{DD} = 1.0 - 5.5\text{V}$ Industrial (I): -40°C to +85°C					
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	TEST CONDITIONS	
Operating Voltage Range		V_{DD}	1.0	5.5	V		
V_{DD} Value to Guarantee RESET/RESET		V_{DDMIN}	1.0		V		
Operating current		I_{DD}		45	60	μA	$V_{DD} = 5.5\text{V}$ (no load)
V _{DD} Trip Point	MCP10X-270		2.55	2.625	2.7	V	
	MCP10X-300		2.85	2.925	3.0		
	MCP10X-315		3.0	3.075	3.15		
	MCP10X-450		4.25	4.375	4.50		
	MCP10X-460		4.35	4.475	4.60		
	MCP10X-475		4.50	4.625	4.75		
	MCP10X-485		4.60	4.725	4.85		
RESET Low Level Output Voltage (MCP100)	MCP100-270				0.4	V	$I_{OL} = 3.2\text{mA}$, $V_{DD} = V_{TRIPMIN}$
	MCP100-300						
	MCP100-315						
	MCP100-450				0.6		
	MCP100-460						
	MCP100-475						
	MCP100-485						
RESET High Level Output Voltage (MCP100)		MCP100-XXX (All V_{TRIP} Points)	V_{OH}	$V_{DD}-0.7$		V	$I_{OH} = 3\text{mA}$, $V_{DD} > V_{TRIPMAX}$
RESET Low Level Output Voltage (MCP101)	MCP101-270				0.4	V	$I_{OL} = 3.2\text{mA}$, $V_{DD} > V_{TRIPMAX}$
	MCP101-300						
	MCP101-315						
	MCP101-450						
	MCP101-460						
	MCP101-475						
	MCP101-485						
RESET High level Output Voltage (MCP101)		MCP101-XXX (All V_{TRIP} Points)	V_{OH}	$V_{DD}-0.7$		V	$I_{OH} = 3\text{mA}$, $V_{DD} = V_{TRIPMIN}$
Threshold Hysteresis		V_{HYS}		50		mV	
V_{DD} Detect to RESET/RESET Inactive		t_{RPUI}	150	350	700	ms	
V_{DD} Detect to RESET/RESET		t_{RPD}		10		μs	V_{DD} ramped from $V_{TRIPMAX} + 250\text{mV}$ down to $V_{TRIPMIN} - 250\text{mV}$

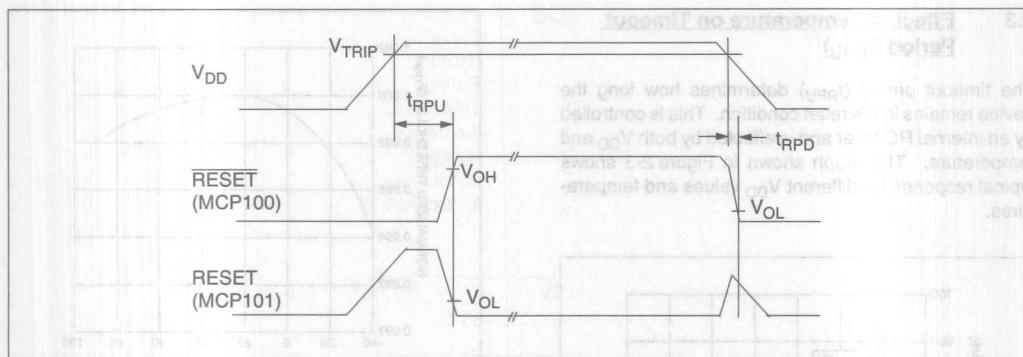


FIGURE 1-1: MCP100/101 Timing Diagram

2.0 APPLICATIONS INFORMATION

2.1 The Need for Supervisory Circuits

For many of today's microcontroller applications, care must be taken to prevent low power conditions that can cause many different system problems. The most common causes are brown-out conditions where the system supply drops below the operating level momentarily, and the second, is when a slowly decaying power supply causes the microcontroller to begin executing instructions without enough voltage to sustain SRAM and producing indeterminate results.

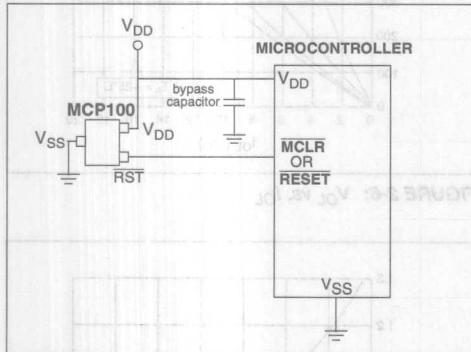


FIGURE 2-1: Typical Application

2.2 Negative Going V_{DD} Transients

Many system designers implementing POR circuits are concerned about the minimum pulse width required to cause a reset. Figure 2-2 shows typical transient duration vs. reset comparator overdrive for which the MCP100/101 will not generate a reset pulse. It shows that the farther below the trip point the transient pulse goes, the duration of the pulse required to cause a reset gets shorter. A 0.1 μ F bypass cap mounted as close as possible to the V_{DD} pin provides additional transient immunity.

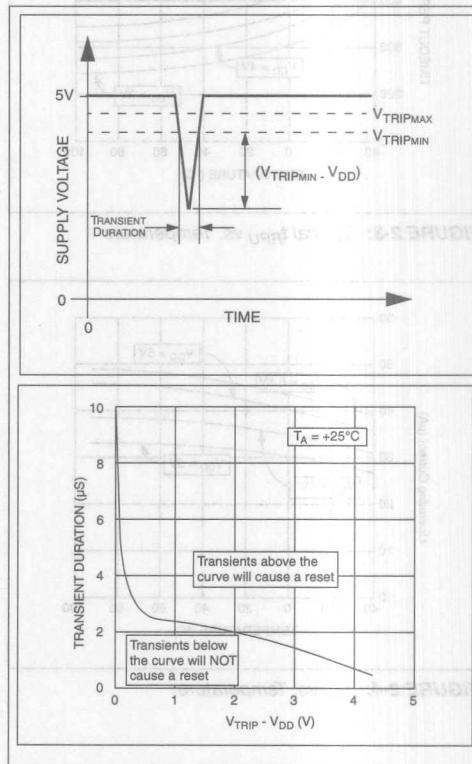


FIGURE 2-2: Typical Transient Response

2.3 EFFECT OF TEMPERATURE ON TIMEOUT PERIOD (t_{RPU})

The timeout period (t_{RPU}) determines how long the device remains in the reset condition. This is controlled by an internal RC timer and is effected by both V_{DD} and temperature. The graph shown in Figure 2-3 shows typical response for different V_{DD} values and temperatures.

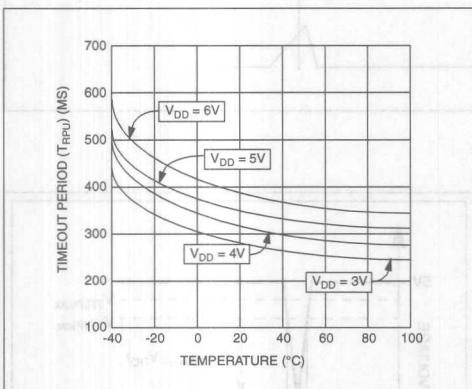


FIGURE 2-3: Typical t_{RPU} vs. Temperature

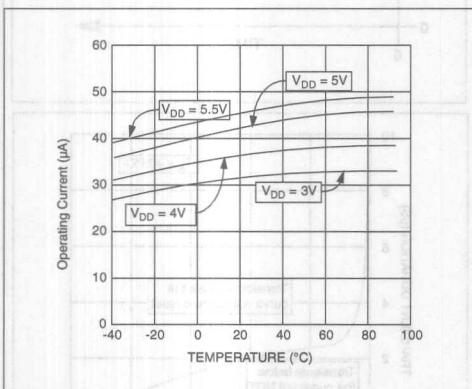
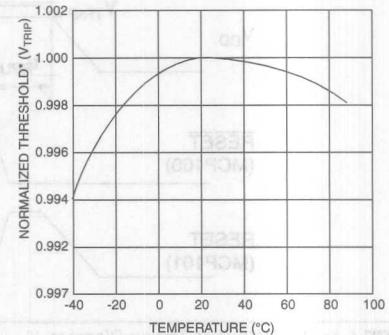


FIGURE 2-4: I_{DD} vs. Temperature



* Multiply value at 25°C by this factor to determine the value at temperature

FIGURE 2-5: Normalized V_{TRIP} vs. Temperature

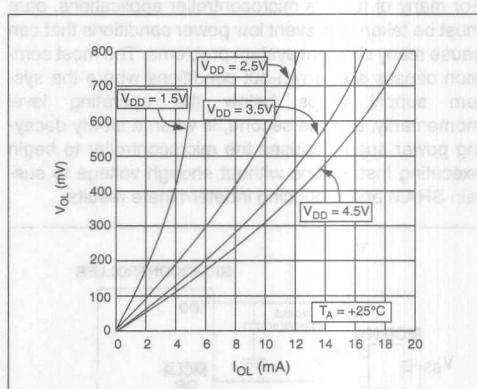
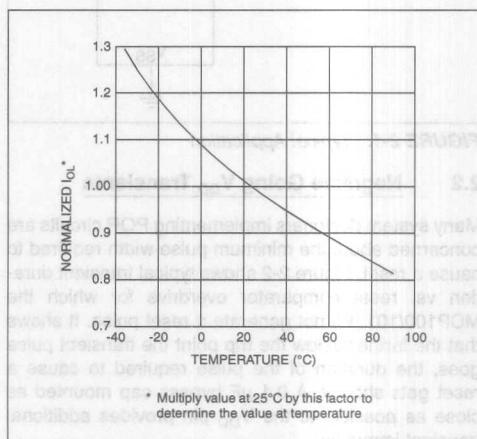


FIGURE 2-6: V_{OL} vs. I_{OL}



* Multiply value at 25°C by this factor to determine the value at temperature

FIGURE 2-7: Normalized I_{OL} vs. Temperature

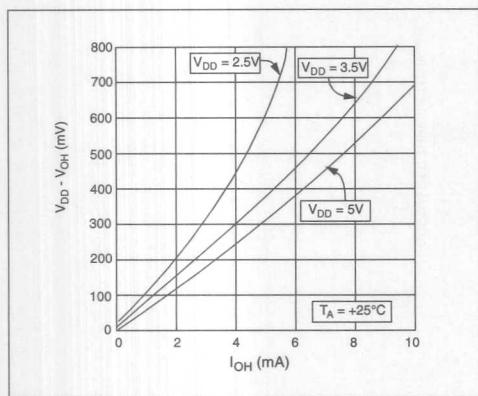


FIGURE 2-8: $V_{DD} - V_{OH}$ vs. I_{OH}

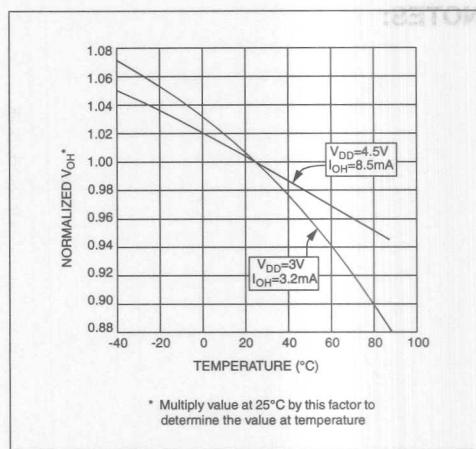
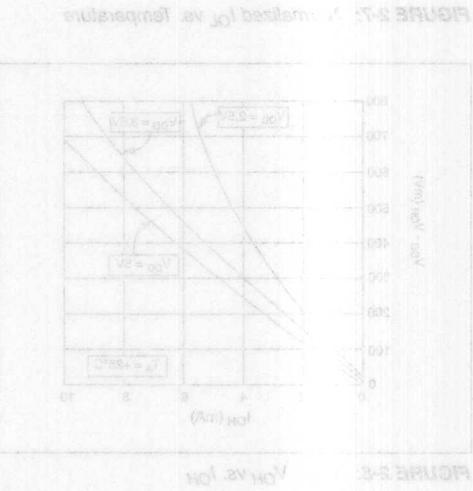
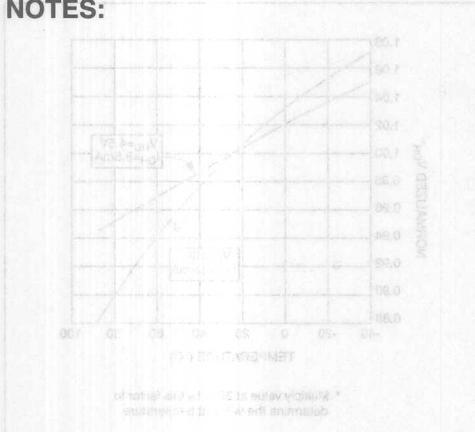


FIGURE 2-9: Normalized V_{OH} vs. Temperature

MCP100/101

NOTES:



DS11187D Rev. B

MCP100/101 PRODUCT IDENTIFICATION SYSTEM

<p>Package:</p> <p>Temperature Range</p> <p>Bondout Option (TO-92 only)</p> <p>RESET/RESET V_{TRIP} Voltage</p> <p>Device:</p>	<p>TO = TO-92 (3-lead) [offered in bags only] TT = SOT-23 (3-lead) [offered in tape & reel only]</p> <p>I = -40°C to +85°C (only offered in I)</p> <p>D = D Bond Option (see bond option chart below) H = H Bond Option</p> <p>270 = 2.55 ≤ V_{TRIP} ≤ 2.70 300 = 2.85 ≤ V_{TRIP} ≤ 3.00 315 = 3.00 ≤ V_{TRIP} ≤ 3.15 450 = 4.25 ≤ V_{TRIP} ≤ 4.50 460 = 4.35 ≤ V_{TRIP} ≤ 4.60 475 = 4.50 ≤ V_{TRIP} ≤ 4.75 485 = 4.60 ≤ V_{TRIP} ≤ 4.85</p> <p>MCP100 = Supervisor circuit with active low $\overline{\text{RESET}}$ output MCP100T = Supervisor circuit with active low $\overline{\text{RESET}}$ output (tape & reel) MCP101 = Supervisor circuit with active high RESET output MCP101T = Supervisor circuit with active high RESET output (tape & reel)</p>		
	<p>Product Identification Options for MCP100/101</p> <table border="1"> <tbody> <tr> <td>TO-92 with 'D' Bondout </td> <td>TO-92 with 'H' Bondout </td> <td>SOT-23-3 </td> </tr> </tbody> </table> <p>Example of Product Identification:</p> <p>For the MCP100 (active low $\overline{\text{RESET}}$ pin) with V_{TRIP} range of 2.55V - 2.70V in TO-92 package with 'H' bond option in industrial temp range, the part number would be: MCP100-270H I/TO.</p>	TO-92 with 'D' Bondout 	TO-92 with 'H' Bondout
TO-92 with 'D' Bondout 	TO-92 with 'H' Bondout 	SOT-23-3 	

3

Datasheets

Sales and Support**Data Sheets**

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277.
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCP100/101

NOTES:



MICROCHIP

MCP120/130

Microcontroller Supervisory Circuit with Open Drain Output

FEATURES

- Holds microcontroller in reset until supply voltage reaches stable operating level
- Resets microcontroller during power loss
- Precision monitoring of 3V, 3.3V, and 5V systems
- 7 voltage trip points available
- Active low RESET pin
- Open drain output
- Internal pullup resistor (5KΩ) for MCP130
- Holds RESET for 350 ms (typical)
- Guaranteed RESET to V_{CC} = 1.0V
- Accuracy of $\pm 125\text{mV}$ for 5V systems and $\pm 75\text{mV}$ for 3V systems over temperature
- 45 μA typical operating current
- Temperature range:
 - Industrial (I): -40°C to +85°C

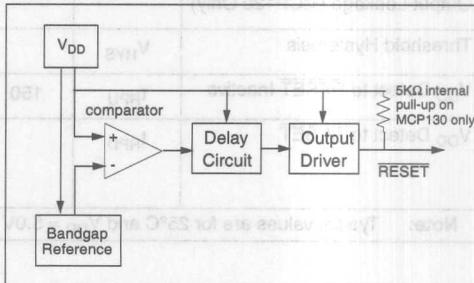
DESCRIPTION

The Microchip Technology Inc. MCP120/130 is a voltage supervisory device designed to keep a microcontroller in reset until the system voltage has reached the proper level and stabilized. It also operates as protection from brown-out conditions when the supply voltage drops below a safe operating level. Both devices are available with a choice of seven different trip voltages and both have open drain outputs. The MCP130 has an internal 5KΩ pullup resistor. Both devices have active low RESET pins. The MCP120/130 will assert the RESET signal whenever the voltage on the V_{DD} pin is below the trip-point voltage.

PACKAGES

 TO-92 with 'D' Bondout MCP120 MCP130	 TO-92 with 'F' Bondout MCP130
<small>MCP120 not offered in this pinout</small>	<small>MCP120 not offered in this pinout</small>
 TO-92 with 'G' Bondout MCP120	 TO-92 with 'H' Bondout MCP130
<small>MCP130 not offered in this pinout</small>	<small>MCP120 MCP130</small>
 SOT-23-3 MCP120 MCP130	 150mil SOIC MCP120 MCP130

BLOCK DIAGRAM



3

Datasheets

MCP120/130

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to $V_{DD} + 1.0V$
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	$\geq 2\text{ kV}$

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

DC AND AC CHARACTERISTICS

All parameters apply at the specified temp and voltage ranges unless otherwise noted.		$V_{DD} = 1.0 - 5.5V$	$V_{DD} = 5.5V$ (no load)				
PARAMETER		SYMBOL	MIN	TYP	MAX	UNITS	TEST CONDITIONS
Operating Voltage Range	V_{DD}		1.0		5.5	V	
V_{DD} Value to Guarantee RESET	V_{DDMIN}		1.0			V	
Operating Current	I_{DD}			45	60	μA	$V_{DD} = 5.5V$ (no load)
V_{DD} Trip Point	MCP1X0-270	V_{TRIP}	2.55	2.625	2.7	V	
	MCP1X0-300		2.85	2.925	3.0		
	MCP1X0-315		3.0	3.075	3.15		
	MCP1X0-450		4.25	4.375	4.50		
	MCP1X0-460		4.35	4.475	4.60		
	MCP1X0-475		4.50	4.625	4.75		
	MCP1X0-485		4.60	4.725	4.85		
RESET Low Level Output Voltage	MCP1X0-270	V_{OL}		0.4	V		$I_{OL} = 3.2\text{mA}$, $V_{DD} = V_{TRIPMIN}$
	MCP1X0-300			0.6	V		$I_{OL} = 8.5\text{mA}$, $V_{DD} = V_{TRIPMIN}$
	MCP1X0-315						
	MCP1X0-450						
	MCP1X0-460						
	MCP1X0-475						
	MCP1X0-485						
RESET High Level Output Voltage (MCP130 Only)	MCP130-XXX (All V_{TRIP} Points)	V_{OH}	$V_{DD}-0.7$		V		$I_{OH} = 50\mu\text{A}$, $V_{DD} > V_{TRIPMAX}$
Pullup Resistor (MCP130 Only)				5		$\text{k}\Omega$	
Output Leakage (MCP120 Only)				1		μA	
Threshold Hysteresis	V_{HYS}			50		mV	
V_{DD} Detect to RESET Inactive	t_{RPU}	150	350	700	ms		
V_{DD} Detect to RESET	t_{RPD}			10	μs		V_{DD} ramped from $V_{TRIPMAX} + 250\text{mV}$ down to $V_{TRIPMIN} - 250\text{mV}$

Note: Typical values are for 25°C and $V_{DD} = 5.0\text{V}$

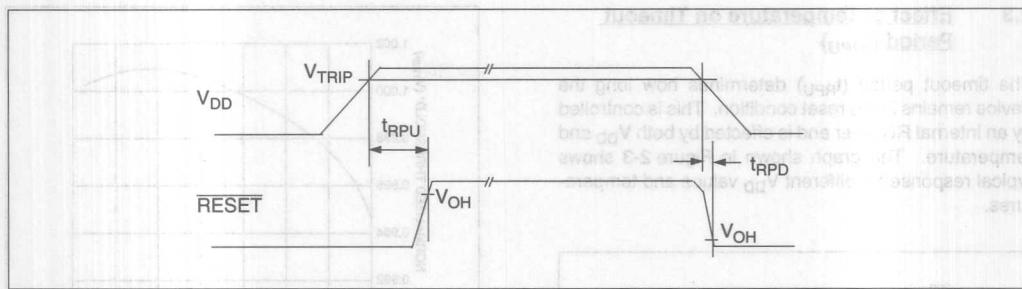


FIGURE 1: MCP120/130 Timing Diagram

2.0 APPLICATIONS INFORMATION

2.1 The Need for Supervisory Circuits

For many of today's microcontroller applications, care must be taken to prevent low power conditions that can cause many different system problems. The most common causes are brown-out conditions where the system supply drops below the operating level momentarily, and the second, is when a slowly decaying power supply causes the microcontroller to begin executing instructions without enough voltage to sustain SRAM and producing indeterminate results.

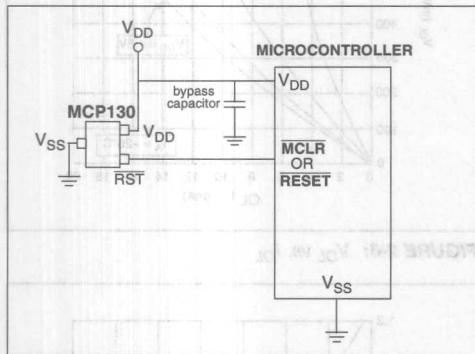


FIGURE 2-1: Typical Application

2.2 Negative Going V_DD Transients

Many system designers implementing POR circuits are concerned about the minimum pulse width required to cause a reset. Figure 2-2 shows typical transient voltage below the trip point ($V_{TRIP} - V_{DD}$) vs. transient duration. It shows that the farther below the trip point the transient pulse goes, the duration of the pulse required to cause a reset gets shorter. A 0.1 μ F bypass cap mounted as close as possible to the V_{DD} pin provides additional transient immunity.

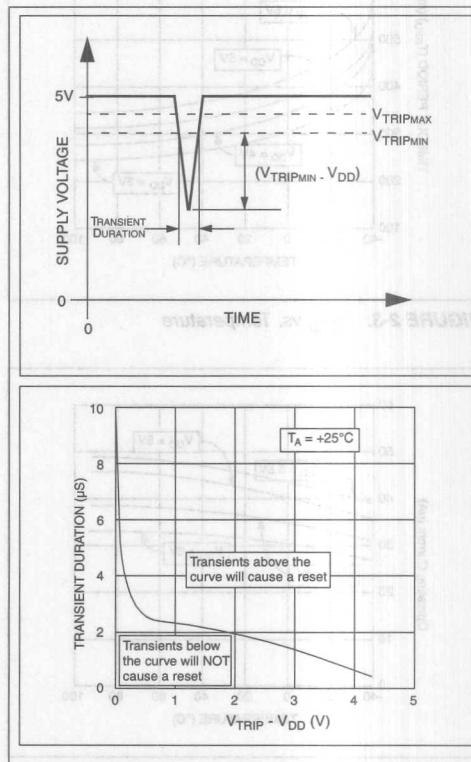


FIGURE 2-2: Typical Transient Response

MCP120/130

2.3 Effect of Temperature on Timeout Period (t_{RPU})

The timeout period (t_{RPU}) determines how long the device remains in the reset condition. This is controlled by an internal RC timer and is effected by both V_{DD} and temperature. The graph shown in Figure 2-3 shows typical response for different V_{DD} values and temperatures.

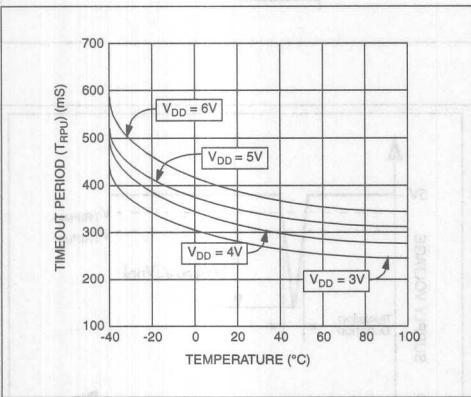


FIGURE 2-3: t_{RPU} vs. Temperature

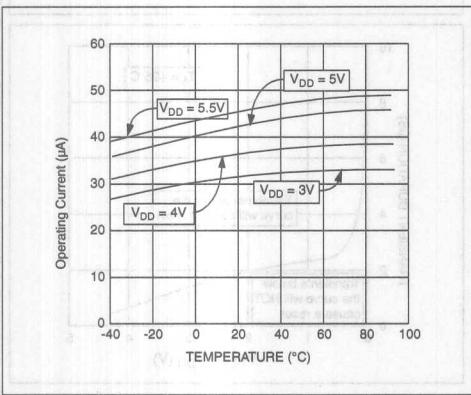


FIGURE 2-4: I_{DD} vs. Temperature

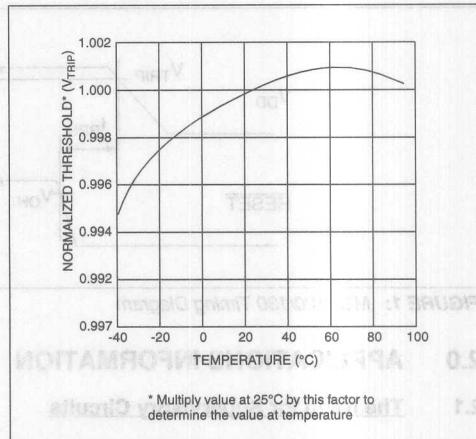


FIGURE 2-5: Normalized V_{TRIP} vs. Temperature

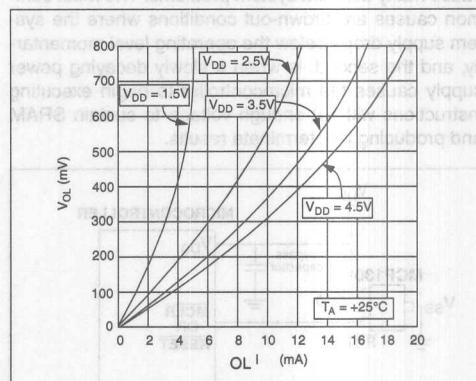


FIGURE 2-6: V_{OL} vs. I_{OL}

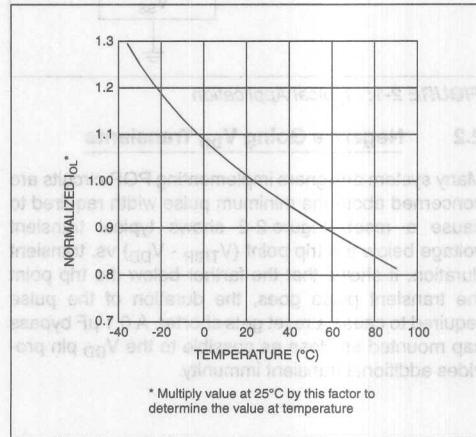
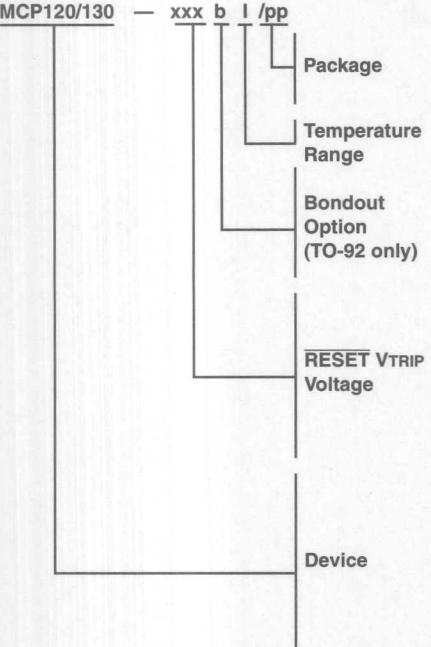
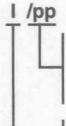
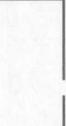
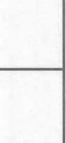
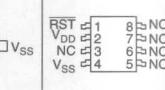
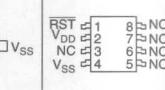
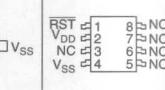


FIGURE 2-7: Normalized I_{OL} vs. Temperature

MCP120/130 PRODUCT IDENTIFICATION SYSTEM

MCP120/130 — xxx b I /pp 	Package 	Temperature Range 	Bondout Option (TO-92 only) 	RESET V_{TRIP} Voltage 	Device 											
	TO = TO-92 (3-lead) [offered in bags only]															
	TT = SOT-23 (3-lead) [offered in tape & reel only]															
	SN = SOIC (8-lead, 150 mil Body)															
	I = -40°C to +85°C (only offered in I)															
	D = D Bond Option (see bond option diagrams)															
270 = $2.55 \leq V_{TRIP} \leq 2.70$ 300 = $2.85 \leq V_{TRIP} \leq 3.00$ 315 = $3.00 \leq V_{TRIP} \leq 3.15$ 450 = $4.25 \leq V_{TRIP} \leq 4.50$ 460 = $4.35 \leq V_{TRIP} \leq 4.60$ 475 = $4.50 \leq V_{TRIP} \leq 4.75$ 485 = $4.60 \leq V_{TRIP} \leq 4.85$																
MCP120 = Supervisor circuit with open drain output. MCP120T = Supervisor circuit with open drain output (tape & reel). MCP130 = Supervisor circuit with open drain output and internal pullup resistor. MCP130T = Supervisor circuit with open drain output and internal pullup resistor (tape & reel).																
Product Identification Options for MCP120/130 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 5px;">TO-92 with 'D' Bondout </td> <td style="text-align: center; padding: 5px;">TO-92 with 'F' Bondout </td> <td style="text-align: center; padding: 5px;">TO-92 with 'G' Bondout </td> <td style="text-align: center; padding: 5px;">TO-92 with 'H' Bondout </td> <td style="text-align: center; padding: 5px;">SOT-23-3 </td> <td style="text-align: center; padding: 5px;">150mil SOIC </td> </tr> <tr> <td style="text-align: center; padding: 5px;">MCP120-XXXD I/TO MCP130-XXXD I/TO</td> <td style="text-align: center; padding: 5px;">MCP130-XXXF I/TO MCP120 not offered in this pinout</td> <td style="text-align: center; padding: 5px;">MCP120-XXXG I/TO MCP130 not offered in this pinout</td> <td style="text-align: center; padding: 5px;">MCP120-XXXH I/TO MCP130-XXXH I/TO</td> <td style="text-align: center; padding: 5px;">MCP120-XXX I/TT MCP130-XXX I/TT</td> <td style="text-align: center; padding: 5px;">MCP120-XXX I/SN MCP130-XXX I/SN</td> </tr> </table>					TO-92 with 'D' Bondout 	TO-92 with 'F' Bondout 	TO-92 with 'G' Bondout 	TO-92 with 'H' Bondout 	SOT-23-3 	150mil SOIC 	MCP120-XXXD I/TO MCP130-XXXD I/TO	MCP130-XXXF I/TO MCP120 not offered in this pinout	MCP120-XXXG I/TO MCP130 not offered in this pinout	MCP120-XXXH I/TO MCP130-XXXH I/TO	MCP120-XXX I/TT MCP130-XXX I/TT	MCP120-XXX I/SN MCP130-XXX I/SN
TO-92 with 'D' Bondout 	TO-92 with 'F' Bondout 	TO-92 with 'G' Bondout 	TO-92 with 'H' Bondout 	SOT-23-3 	150mil SOIC 											
MCP120-XXXD I/TO MCP130-XXXD I/TO	MCP130-XXXF I/TO MCP120 not offered in this pinout	MCP120-XXXG I/TO MCP130 not offered in this pinout	MCP120-XXXH I/TO MCP130-XXXH I/TO	MCP120-XXX I/TT MCP130-XXX I/TT	MCP120-XXX I/SN MCP130-XXX I/SN											
Example of Product Identification: For the MCP130 (open drain output with internal pull up resistor) with V_{TRIP} range of 2.55V - 2.70V in TO-92 package with 'F' bond option in industrial temp range, the part number would be: MCP130-270F I/TO.																

Sales and Support

Data Sheets

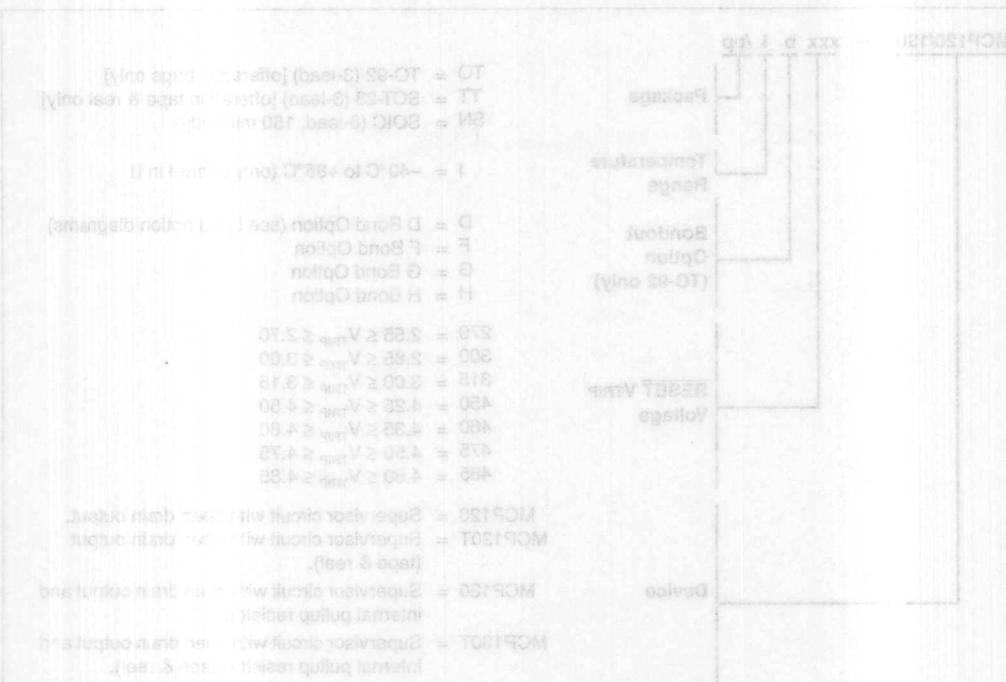
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

NOTES:**MCP15035 PRODUCTS IDENTIFICATION SYSTEM**

Product Identification Options for MCP15035

Product ID	Range Selection	Range Options	Range Options	Configuration Options
1000 0000 0000 0000	Range 0	D	E	H
1000 0000 0001 0000	Range 1	D	E	I
1000 0000 0010 0000	Range 2	D	F	J
1000 0000 0011 0000	Range 3	D	G	K

Example of Product Identification:

For the MCP15035 with the following product ID:
 MSB: 10
 Range: 0
 Bonding: D
 Configuration: H
 Final Product: MCP15035-AH

Design Guidelines

- Please refer to the following documents for detailed information on design guidelines:
 - 1. [Microchip Technology Design Guide](#)
 - 2. [Microchip Technology Design Guide](#)
 - 3. [Microchip Technology Design Guide](#)
- Please refer to the following documents for detailed information on design guidelines:
 - 1. [Microchip Technology Design Guide](#)
 - 2. [Microchip Technology Design Guide](#)
 - 3. [Microchip Technology Design Guide](#)

Microcontroller Supervisory Circuit with Push-Pull Output

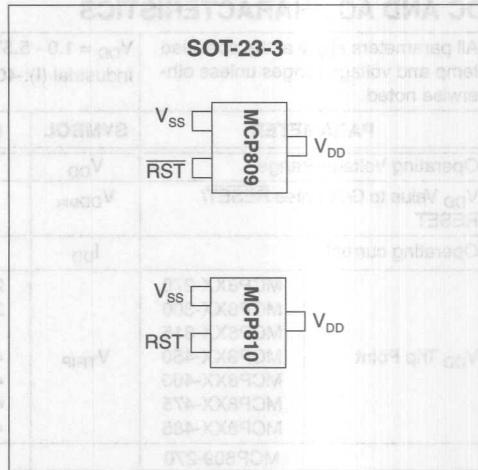
FEATURES

- Holds microcontroller in reset until supply voltage reaches stable operating level
- Resets microcontroller during power loss
- Precision monitoring of 3V, 3.3V, and 5V systems
- 7 voltage trip points available
- Active low $\overline{\text{RESET}}$ pin (MCP809) or active high RESET (MCP810)
- Push-pull output
- Holds $\overline{\text{RESET}}/\text{RESET}$ for 350 ms (typical)
- Guaranteed $\overline{\text{RESET}}/\text{RESET}$ to $V_{DD} = 1.0V$
- Accuracy of $\pm 125mV$ for 5V systems and $\pm 75mV$ for 3V systems over temperature
- 45 μA typical operating current
- Temperature range:
 - Industrial (I): -40°C to +85°C

DESCRIPTION

The Microchip Technology Inc. MCP809/810 is a voltage supervisory device designed to keep a microcontroller in reset until the system voltage has reached the proper level and stabilized. It also operates as protection from brown-out conditions when the supply voltage drops below a safe operating level. Both devices are available with a choice of seven different trip voltages and both have push-pull outputs. The MCP809 has a low active $\overline{\text{RESET}}$ pin and the MCP810 has a high active RESET pin. The MCP809/810 will assert the $\overline{\text{RESET}}/\text{RESET}$ signal whenever the voltage on the V_{DD} pin is below the trip-point voltage.

PACKAGES



MCP809/810

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to $V_{DD} + 1.0V$
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	$\geq 2\text{ kV}$

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

DC AND AC CHARACTERISTICS

All parameters apply at the specified temp and voltage ranges unless otherwise noted.		$V_{DD} = 1.0 - 5.5\text{V}$ Industrial (I): -40°C to +85°C				
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	TEST CONDITIONS
Operating Voltage Range	V_{DD}	1.0	1.2	5.5	5.5	$V_{DD} = 5.5\text{V}$ (no load)
V_{DD} Value to Guarantee RESET/RESET	V_{DDMIN}	1.0			V	
Operating current	I_{DD}		45	60	μA	$V_{DD} = 5.5\text{V}$ (no load)
V_{DD} Trip Point	MCP8XX-270 MCP8XX-300 MCP8XX-315 MCP8XX-450 MCP8XX-460 MCP8XX-475 MCP8XX-485	V_{TRIP}	2.55 2.85 3.0 4.25 4.35 4.50 4.60	2.625 2.925 3.075 4.375 4.475 4.625 4.725	2.7 3.0 3.15 4.50 4.60 4.75 4.85	V
RESET Low Level Output Voltage (MCP809)	MCP809-270 MCP809-300 MCP809-315 MCP809-450 MCP809-460 MCP809-475 MCP809-485	V_{OL}			0.4	V
RESET High Level Output Voltage (MCP809)	MCP809-XXX (All V_{TRIP} Points)	V_{OH}	$V_{DD}-0.7$		V	$I_{OH} = 3\text{mA}, V_{DD} > V_{TRIPMAX}$
RESET Low Level Output Voltage (MCP810)	MCP810-270 MCP810-300 MCP810-315 MCP810-450 MCP810-460 MCP810-475 MCP810-485	V_{OL}			0.4	V
RESET High level Output Voltage (MCP810)	MCP810-XXX (All V_{TRIP} Points)	V_{OH}	$V_{DD}-0.7$		V	$I_{OH} = 3\text{mA}, V_{DD} = V_{TRIPMIN}$
Threshold Hysteresis	V_{HYS}		50		mV	
V_{DD} Detect to RESET/RESET Inactive	t_{RPU}	150	350	700	ms	
V_{DD} Detect to RESET/RESET	t_{RPD}		10		μs	V_{DD} ramped from $V_{TRIPMAX} + 250\text{mV}$ down to $V_{TRIPMIN} - 250\text{mV}$

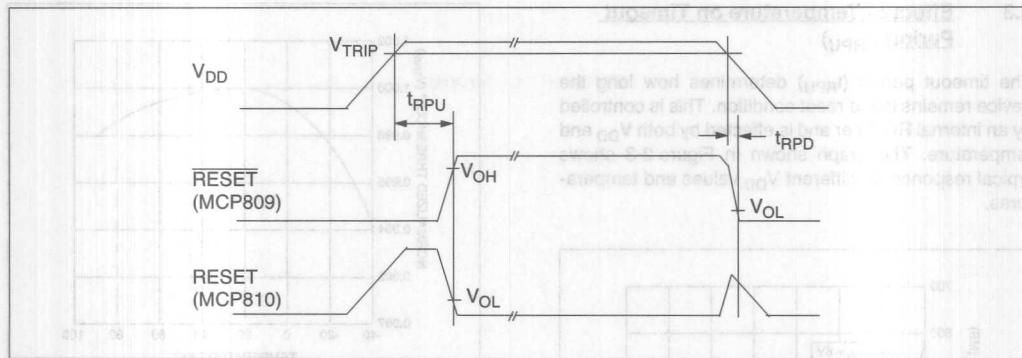


FIGURE 1-1: MCP809/810 Timing Diagram

2.0 APPLICATIONS INFORMATION

2.1 The Need for Supervisory Circuits

For many of today's microcontroller applications, care must be taken to prevent low power conditions that can cause many different system problems. The most common causes are brown-out conditions where the system supply drops below the operating level momentarily, and the second, is when a slowly decaying power supply causes the microcontroller to begin executing instructions without enough voltage to sustain SRAM and producing indeterminate results.

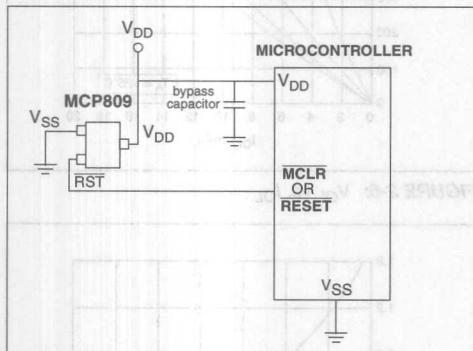


FIGURE 2-1: Typical Application

2.2 Negative Going V_{DD} Transients

Many system designers implementing POR circuits are concerned about the minimum pulse width required to cause a reset. Figure 2-2 shows typical transient duration vs. reset comparator overdrive for which the MCP809/810 will not generate a reset pulse. It shows that the farther below the trip point the transient pulse goes, the duration of the pulse required to cause a reset gets shorter. A 0.1 μ F bypass cap mounted as close as possible to the V_{DD} pin provides additional transient immunity.

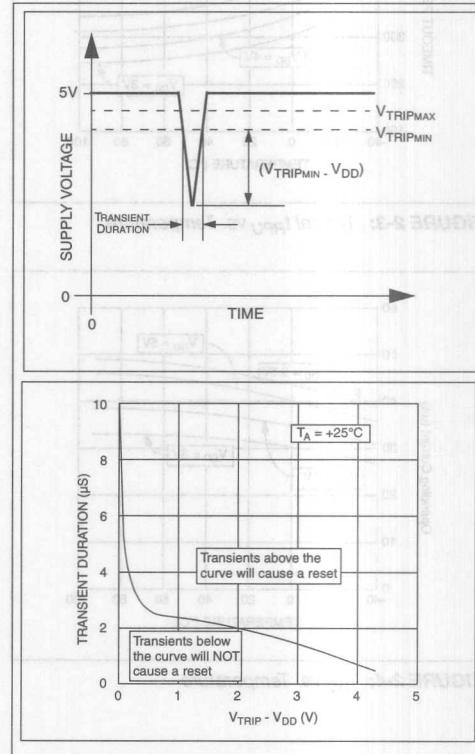


FIGURE 2-2: Typical Transient Response

2.3 Effect of Temperature on Timeout Period (t_{RPU})

The timeout period (t_{RPU}) determines how long the device remains in the reset condition. This is controlled by an internal RC timer and is effected by both V_{DD} and temperature. The graph shown in Figure 2-3 shows typical response for different V_{DD} values and temperatures.

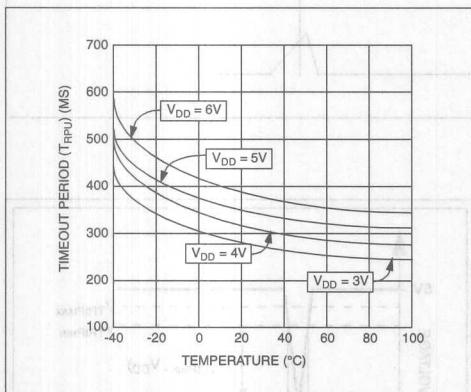


FIGURE 2-3: Typical t_{RPU} vs. Temperature

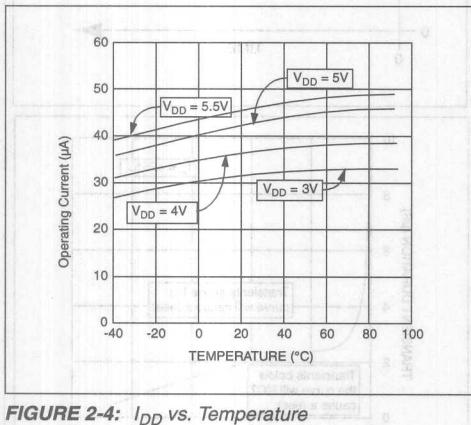


FIGURE 2-4: I_{DD} vs. Temperature

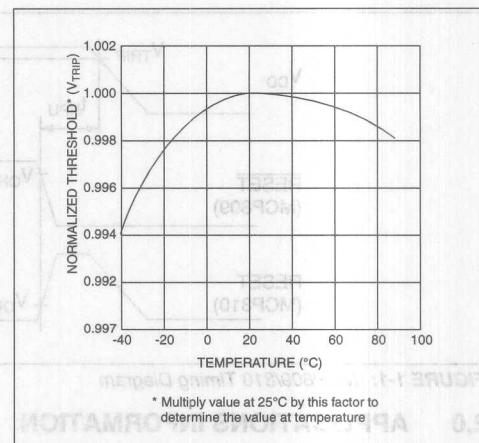


FIGURE 2-5: Normalized V_{Ttrip} vs. Temperature

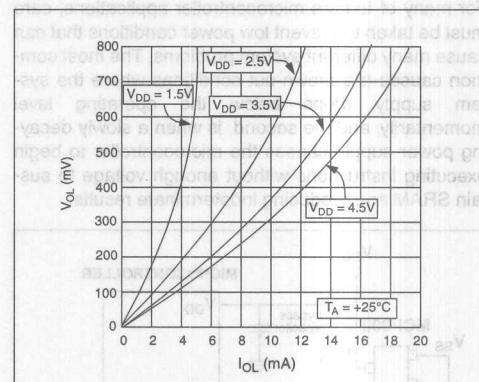


FIGURE 2-6: V_{OL} vs. I_{OL}

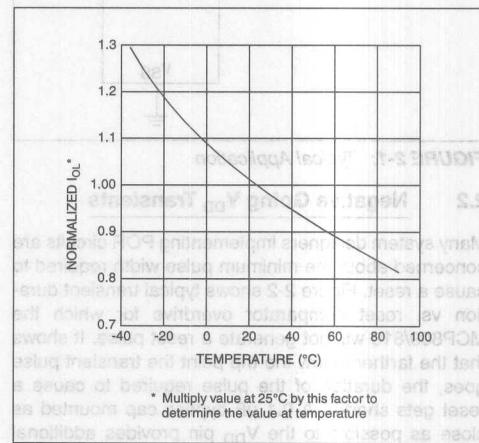


FIGURE 2-7: Normalized I_{OL} vs. Temperature

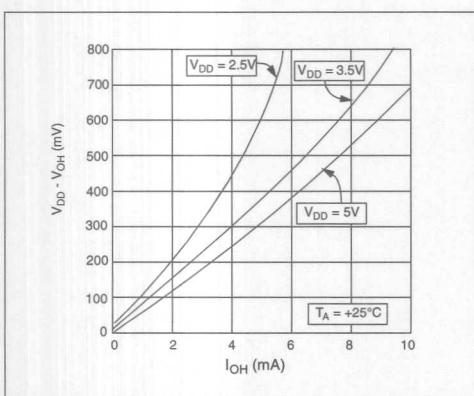


FIGURE 2-8: $V_{DD} - V_{OH}$ vs. I_{OH}

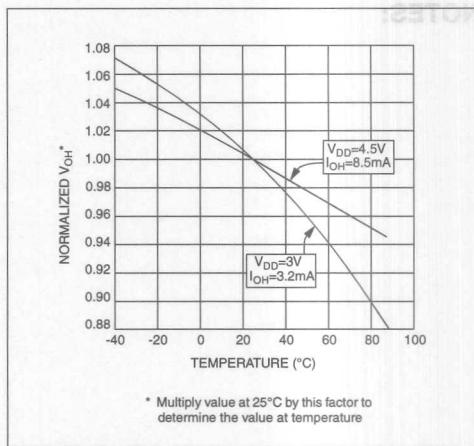
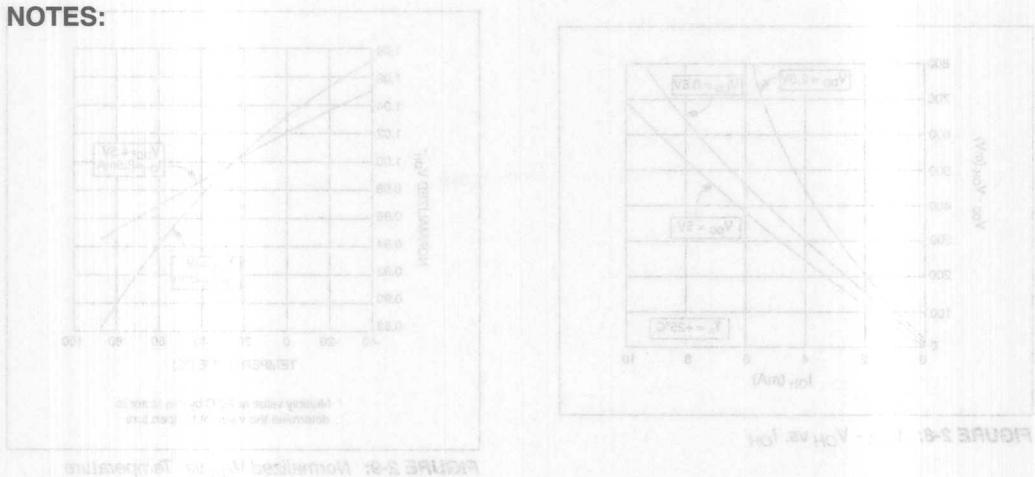


FIGURE 2-9: Normalized V_{OH} vs. Temperature

MCP809/810

NOTES:



Microchip Technology Inc.

MCP809/810 PRODUCT IDENTIFICATION SYSTEM

MCP809/810T — xxx I /pp	Package:	TT = SOT-23 (3-lead) [offered in tape & reel only]
	Temperature Range	I = -40°C to +85°C (only offered in I)
	RESET/RESET V_{TRIP} Voltage	270 = $2.55 \leq V_{TRIP} \leq 2.70$ 300 = $2.85 \leq V_{TRIP} \leq 3.00$ 315 = $3.00 \leq V_{TRIP} \leq 3.15$ 450 = $4.25 \leq V_{TRIP} \leq 4.50$ 460 = $4.35 \leq V_{TRIP} \leq 4.60$ 475 = $4.50 \leq V_{TRIP} \leq 4.75$ 485 = $4.60 \leq V_{TRIP} \leq 4.85$
	Device:	MCP809T = Supervisor circuit with active low RESET output (tape & reel) MCP810T = Supervisor circuit with active high RESET output (tape & reel)

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCDONALD'S

Stand-Alone CAN Controller with SPI™ Interface

FEATURES

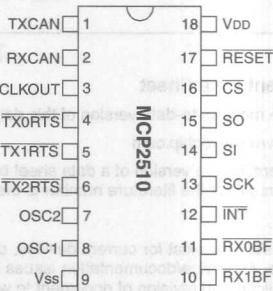
- Implements Full CAN V2.0A and V2.0B at 1 Mb/s
 - 0 - 8 byte message length
 - Standard and extended data frames
 - Programmable bit rate up to 1 Mb/s
 - Support for remote frames
 - Two receive buffers with prioritized message storage
 - Six full acceptance filters
 - Two full acceptance filter masks
 - Three transmit buffers with prioritization and abort features
 - Loop-back mode for self test operation
- Hardware Features
 - High Speed SPI Interface (5 MHz at 4.5V I temp)
 - Supports SPI™ modes 0,0 and 1,1
 - Clock out pin with programmable prescaler
 - Interrupt output pin with selectable enables
 - 'Buffer full' output pins configurable as interrupt pins for each receive buffer or as general purpose digital outputs
 - 'Request to Send' input pins configurable as control pins to request immediate message transmission for each transmit buffer or as general purpose digital inputs
 - Low Power Sleep mode
- Low power CMOS technology
 - Operates from 3.0V to 5.5V
 - 5 mA active current typical
 - 10 µA standby current typical at 5.5V
- 18-pin PDIP/SOIC and 20-pin TSSOP packages
- Temperature ranges supported:
 - Industrial (I): -40°C to +85°C
 - Extended (E): -40°C to +125°C

DESCRIPTION

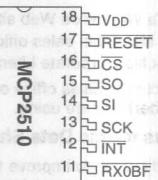
The Microchip Technology Inc. MCP2510 is a Full Controller Area Network (CAN) protocol controller implementing CAN specification V2.0 A/B. It supports CAN 1.2, CAN 2.0A, CAN 2.0B Passive, and CAN 2.0B Active versions of the protocol, and is capable of transmitting and receiving standard and extended messages. It is also capable of both acceptance filtering and message management. It includes three transmit buffers and two receive buffers that reduce the amount of microcontroller (MCU) management required. The MCU communication is implemented via an industry standard Serial Peripheral Interface (SPI) with data rates up to 5Mb/s.

PACKAGE TYPES

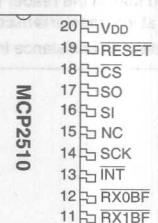
18 LEAD PDIP



18 LEAD SOIC



20 LEAD TSSOP



3

Datasheets

SPI is a trademark of Motorola Inc.

MCP2510

Table of Contents

1.0	Device Functionality	3
2.0	Can Message Frames	7
3.0	Message Transmission	15
4.0	Message Reception	21
5.0	Bit Timing	34
6.0	Error Detection	40
7.0	Interrupts	44
8.0	Oscillator	48
9.0	Modes of Operation	50
10.0	Register Map	53
11.0	SPI Interface	55
12.0	Electrical Characteristics	59
13.0	Packaging Information	63
On-Line Support		67
Reader Response		68
Product Identification System		69
Index		71
List Of Figures		73
List Of Tables		73
List Of Registers		73
Worldwide Sales and Service		74

PACKAGE TYPES

18 LEAD PLD

18 LEAD QFP

18 LEAD TQFP

18 LEAD BGA

18 LEAD QFN

18 LEAD LQFP

18 LEAD QVFBGA

1.0 DEVICE FUNCTIONALITY

1.1 Overview

The MCP2510 is a stand-alone CAN controller developed to simplify applications that require interfacing with a CAN bus. A simple block diagram of the MCP2510 is shown in Figure 1-1. The device consists of three main blocks:

1. the CAN protocol engine,
2. the control logic and SRAM registers that are used to configure the device and its operation, and
3. the SPI protocol block.

A typical system implementation using the device is shown in Figure 1-2.

The CAN protocol engine handles all functions for receiving and transmitting messages on the bus. Messages are transmitted by first loading the appropriate message buffer and control registers. Transmission is initiated by using control register bits, via the SPI interface, or by using the transmit enable pins. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is

checked for errors and then matched against the user defined filters to see if it should be moved into one of the two receive buffers.

The MCU interfaces to the device via the SPI interface. Writing to and reading from all registers is done using standard SPI read and write commands.

Interrupt pins are provided to allow greater system flexibility. There is one multi-purpose interrupt pin as well as specific interrupt pins for each of the receive registers that can be used to indicate when a valid message has been received and loaded into one of the receive buffers. Use of the specific interrupt pins is optional, and the general purpose interrupt pin as well as status registers (accessed via the SPI interface) can also be used to determine when a valid message has been received.

There are also three pins available to initiate immediate transmission of a message that has been loaded into one of the three transmit registers. Use of these pins is optional and initiating message transmission can also be done by utilizing control registers accessed via the SPI interface.

Table 1-1 gives a complete list of all of the pins on the MCP2510.

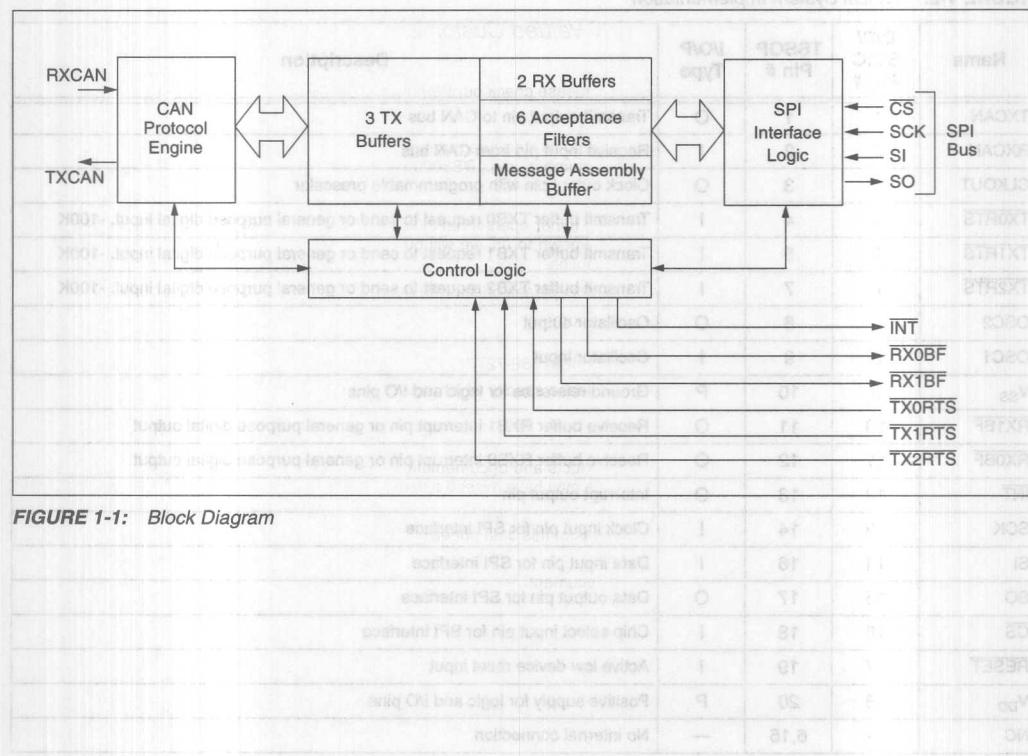


FIGURE 1-1: Block Diagram

MCP2510

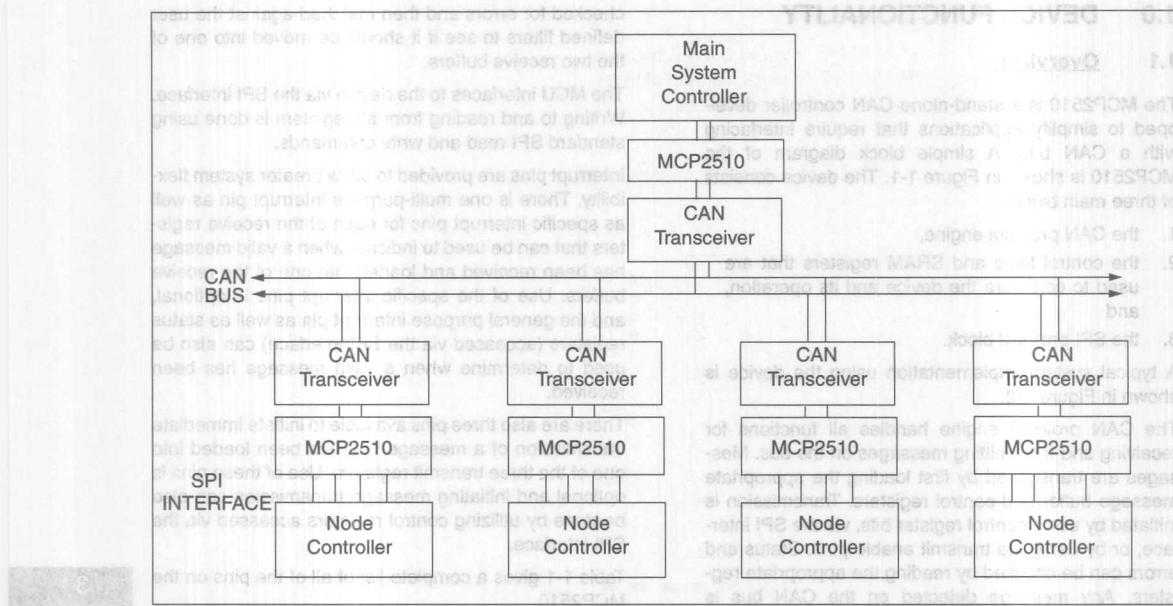


FIGURE 1-2: Typical System Implementation

Name	DIP/ SOIC Pin #	TSSOP Pin #	I/O/P Type	Description	
				SPI	CAN
TXCAN	1	1	O	Transmit output pin to CAN bus	XTC
RXCAN	2	2	I	Receive input pin from CAN bus	XRC
CLKOUT	3	3	O	Clock output pin with programmable prescaler	XCKOUT
TX0RTS	4	4	I	Transmit buffer TXB0 request to send or general purpose digital input. -100K	XRTS0
TX1RTS	5	5	I	Transmit buffer TXB1 request to send or general purpose digital input. -100K	XRTS1
TX2RTS	6	7	I	Transmit buffer TXB2 request to send or general purpose digital input. -100K	XRTS2
OSC2	7	8	O	Oscillator output	XOSC2
OSC1	8	9	I	Oscillator input	XOSC1
V _{ss}	9	10	P	Ground reference for logic and I/O pins	XVSS
RX1BF	10	11	O	Receive buffer RXB1 interrupt pin or general purpose digital output	XRX1BF
RX0BF	11	12	O	Receive buffer RXB0 interrupt pin or general purpose digital output	XRX0BF
INT	12	13	O	Interrupt output pin	XINT
SCK	13	14	I	Clock input pin for SPI interface	XSPICLK
SI	14	16	I	Data input pin for SPI interface	XSPIDI
SO	15	17	O	Data output pin for SPI interface	XSPIDQ
CS	16	18	I	Chip select input pin for SPI interface	XSPICS
RESET	17	19	I	Active low device reset input	XRESET
V _{DD}	18	20	P	Positive supply for logic and I/O pins	XVDD
NC	—	6,15	—	No internal connection	XNC

Note: Type Identification: I=Input; O=Output; P=Power

TABLE 1-1: Pin Descriptions

1.2 Transmit/Receive Buffers

The MCP2510 has three transmit and two receive buffers, two acceptance masks (one for each receive buffer), and a total of six acceptance filters. Figure 1-3 is a block diagram of these buffers and their connection to the protocol engine.

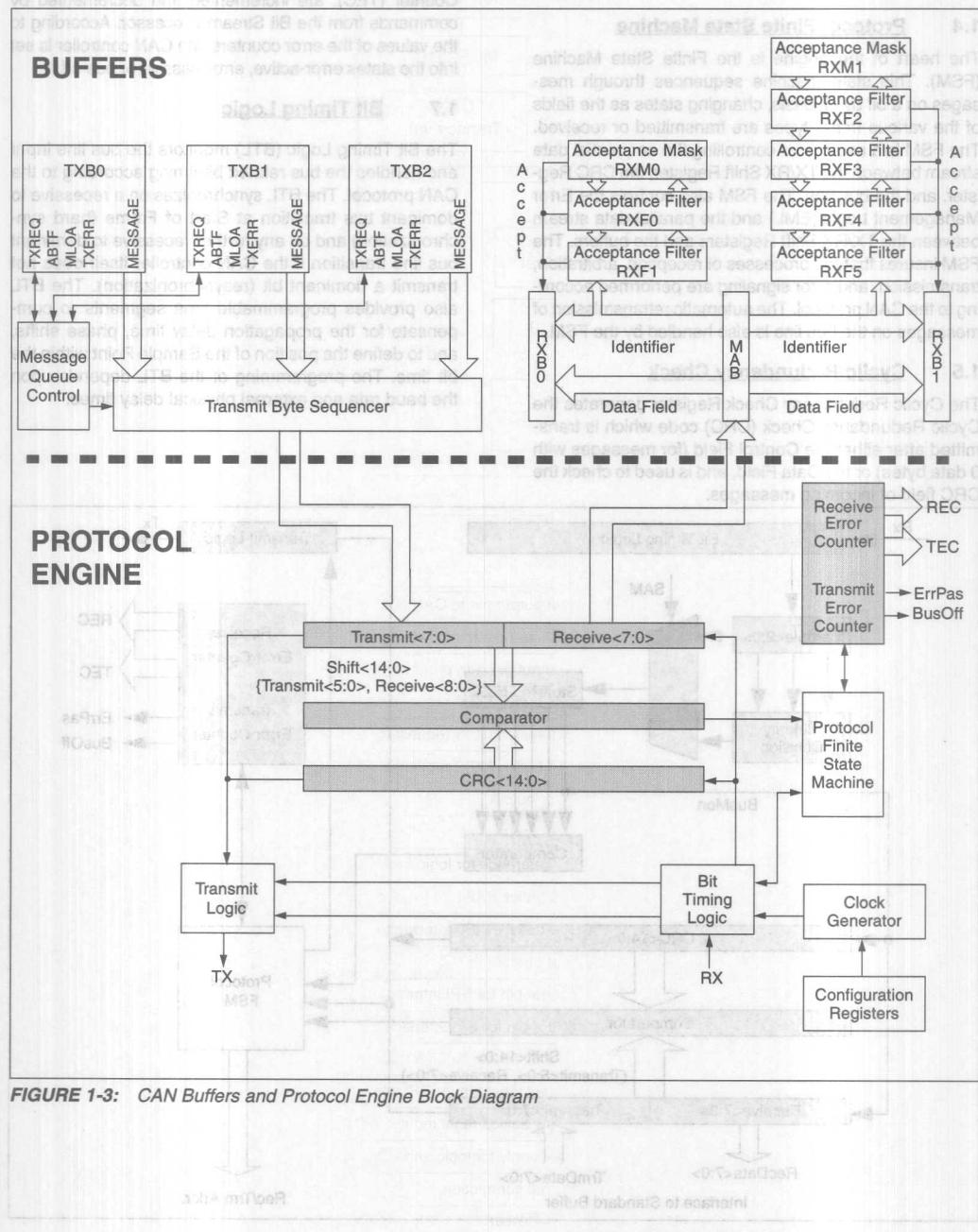


FIGURE 1-3: CAN Buffers and Protocol Engine Block Diagram

The CAN protocol engine combines several functional blocks, shown in Figure 1-4. These blocks and their functions are described below.

1.4 Protocol Finite State Machine

The heart of the engine is the Finite State Machine (FSM). This state machine sequences through messages on a bit by bit basis, changing states as the fields of the various frame types are transmitted or received. The FSM is a sequencer controlling the sequential data stream between the TX/RX Shift Register, the CRC Register, and the bus line. The FSM also controls the Error Management Logic (EML) and the parallel data stream between the TX/RX Shift Registers and the buffers. The FSM insures that the processes of reception, arbitration, transmission, and error signaling are performed according to the CAN protocol. The automatic retransmission of messages on the bus line is also handled by the FSM.

1.5 Cyclic Redundancy Check

The Cyclic Redundancy Check Register generates the Cyclic Redundancy Check (CRC) code which is transmitted after either the Control Field (for messages with 0 data bytes) or the Data Field, and is used to check the CRC field of incoming messages.

The Error Management Logic is responsible for the fault confinement of the CAN device. Its two counters, the Receive Error Counter (REC) and the Transmit Error Counter (TEC), are incremented and decremented by commands from the Bit Stream Processor. According to the values of the error counters, the CAN controller is set into the states error-active, error-passive or bus-off.

1.7 Bit Timing Logic

The Bit Timing Logic (BTL) monitors the bus line input and handles the bus related bit timing according to the CAN protocol. The BTL synchronizes on a recessive to dominant bus transition at Start of Frame (hard synchronization) and on any further recessive to dominant bus line transition if the CAN controller itself does not transmit a dominant bit (resynchronization). The BTL also provides programmable time segments to compensate for the propagation delay time, phase shifts, and to define the position of the Sample Point within the bit time. The programming of the BTL depends upon the baud rate and external physical delay times.

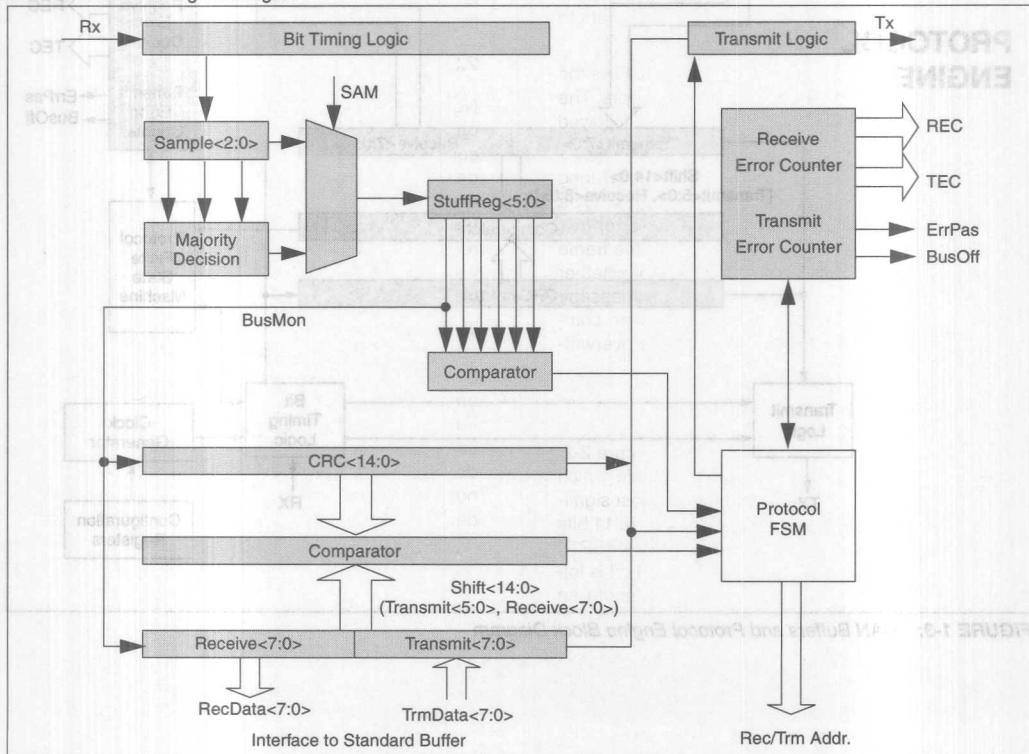


FIGURE 1-4: CAN Protocol Engine Block Diagram

2.0 CAN MESSAGE FRAMES

The MCP2510 supports Standard Data Frames, Extended Data Frames, and Remote Frames (Standard and Extended) as defined in the CAN 2.0B specification.

2.1 Standard Data Frame

The CAN Standard Data Frame is shown in Figure 2-1. In common with all other frames, the frame begins with a Start Of Frame (SOF) bit, which is of the dominant state, which allows hard synchronization of all nodes.

The SOF is followed by the arbitration field, consisting of 12 bits; the 11-bit Identifier and the Remote Transmission Request (RTR) bit. The RTR bit is used to distinguish a data frame (RTR bit dominant) from a remote frame (RTR bit recessive).

Following the arbitration field is the control field, consisting of six bits. The first bit of this field is the Identifier Extension (IDE) bit which must be dominant to specify a standard frame. The following bit, Reserved Bit Zero (RB0), is reserved and is defined to be a dominant bit by the can protocol. the remaining four bits of the control field are the Data Length Code (DLC) which specifies the number of bytes of data contained in the message.

After the control field is the data field, which contains any data bytes that are being sent, and is of the length defined by the DLC above (0-8 bytes).

The Cyclic Redundancy Check (CRC) Field follows the data field and is used to detect transmission errors. The CRC Field consists of a 15-bit CRC sequence, followed by the recessive CRC Delimiter bit.

The final field is the two-bit acknowledge field. During the ACK Slot bit, the transmitting node sends out a recessive bit. Any node that has received an error free frame acknowledges the correct reception of the frame by sending back a dominant bit (regardless of whether the node is configured to accept that specific message or not). The recessive acknowledge delimiter completes the acknowledge field and may not be overwritten by a dominant bit.

2.2 Extended Data Frame

In the Extended CAN Data Frame, shown in Figure 2-2, the SOF bit is followed by the arbitration field which consists of 32 bits. The first 11 bits are the most significant bits (Base-ID) of the 29-bit identifier. These 11 bits are followed by the Substitute Remote Request (SRR) bit which is defined to be recessive. The SRR bit is followed by the IDE bit which is recessive to denote an extended CAN frame.

It should be noted that if arbitration remains unresolved after transmission of the first 11 bits of the identifier, and one of the nodes involved in the arbitration is sending a standard CAN frame (11-bit identifier), then the standard CAN frame will win arbitration due to the assertion of a dominant IDE bit. Also, the SRR bit in an extended CAN frame must be recessive to allow the assertion of a dominant RTR bit by a node that is sending a standard CAN remote frame.

The SRR and IDE bits are followed by the remaining 18 bits of the identifier (Extended ID) and the remote transmission request bit.

To enable standard and extended frames to be sent across a shared network, it is necessary to split the 29-bit extended message identifier into 11-bit (most significant) and 18-bit (least significant) sections. This split ensures that the IDE bit can remain at the same bit position in both standard and extended frames.

Following the arbitration field is the six-bit control field, the first two bits of this field are reserved and must be dominant. the remaining four bits of the control field are the Data Length Code (DLC) which specifies the number of data bytes contained in the message.

The remaining portion of the frame (data field, CRC field, acknowledge field, end of frame and Intermission) is constructed in the same way as for a standard data frame (see Section 2.1).

2.3 Remote Frame

Normally, data transmission is performed on an autonomous basis by the data source node (e.g. a sensor sending out a data frame). It is possible, however, for a destination node to request data from the source. To accomplish this, the destination node sends a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node will then send a data frame in response to the remote frame request.

There are two differences between a remote frame (shown in Figure 2-3) and a data frame. First, the RTR bit is at the recessive state, and second, there is no data field. In the event of a data frame and a remote frame with the same identifier being transmitted at the same time, the data frame wins arbitration due to the dominant RTR bit following the identifier. In this way, the node that transmitted the remote frame receives the desired data immediately.

2.4 Error Frame

An Error Frame is generated by any node that detects a bus error. An error frame, shown in Figure 2-4, consists of two fields, an error flag field followed by an error delimiter field. There are two types of error flag fields. Which type of error flag field is sent depends upon the error status of the node that detects and generates the error flag field.

If an error-active node detects a bus error then the node interrupts transmission of the current message by generating an active error flag. The active error flag is composed of six consecutive dominant bits. This bit sequence actively violates the bit stuffing rule. All other stations recognize the resulting bit stuffing error and in turn generate error frames themselves, called error echo flags. The error flag field, therefore, consists of between six and twelve consecutive dominant bits (generated by one or more nodes). The error delimiter field completes the error frame. After completion of the error frame, bus activity returns to normal and the interrupted node attempts to resend the aborted message.

If an error-passive node detects a bus error then the node transmits an error-passive flag followed by the error delimiter field. The error-passive flag consists of six consecutive recessive bits, and the error frame for an error-passive node consists of 14 recessive bits. From this, it follows that unless the bus error is detected by the node that is actually transmitting, the transmission of an error frame by an error-passive node will not affect any other node on the network. If the transmitting node generates an error-passive flag then this will cause other nodes to generate error frames due to the resulting bit stuffing violation. After transmission of an error frame, an error-passive node must wait for six consecutive recessive bits on the bus before attempting to rejoin bus communications.

The error delimiter consists of eight recessive bits and allows the bus nodes to restart bus communications cleanly after an error has occurred.

2.5 Overload Frame

An Overload Frame, shown in Figure 2-5, has the same format as an active error frame. An overload frame, however can only be generated during an interframe space. In this way an overload frame can be differentiated from an error frame (an error frame is sent during the transmission of a message). The overload frame consists of two fields, an overload flag followed by an overload delimiter. The overload flag consists of six dominant bits followed by overload flags generated by other nodes (and, as for an active error flag, giving a maximum of twelve dominant bits). The overload delimiter consists of eight recessive bits. An overload frame can be generated by a node as a result of two conditions. First, the node detects a dominant bit during the interframe space which is an illegal condition. Second, due to internal conditions the node is not yet able to start reception of the next message. A node may generate a maximum of two sequential overload frames to delay the start of the next message.

2.6 Interframe Spacing

The Interframe Space separates a preceding frame (of any type) from a subsequent data or remote frame. The interframe space is composed of at least three recessive bits called the Intermission. This is provided to allow nodes time for internal processing before the start of the next message frame. After the intermission, the bus line remains in the recessive state (bus idle) until the next transmission starts.

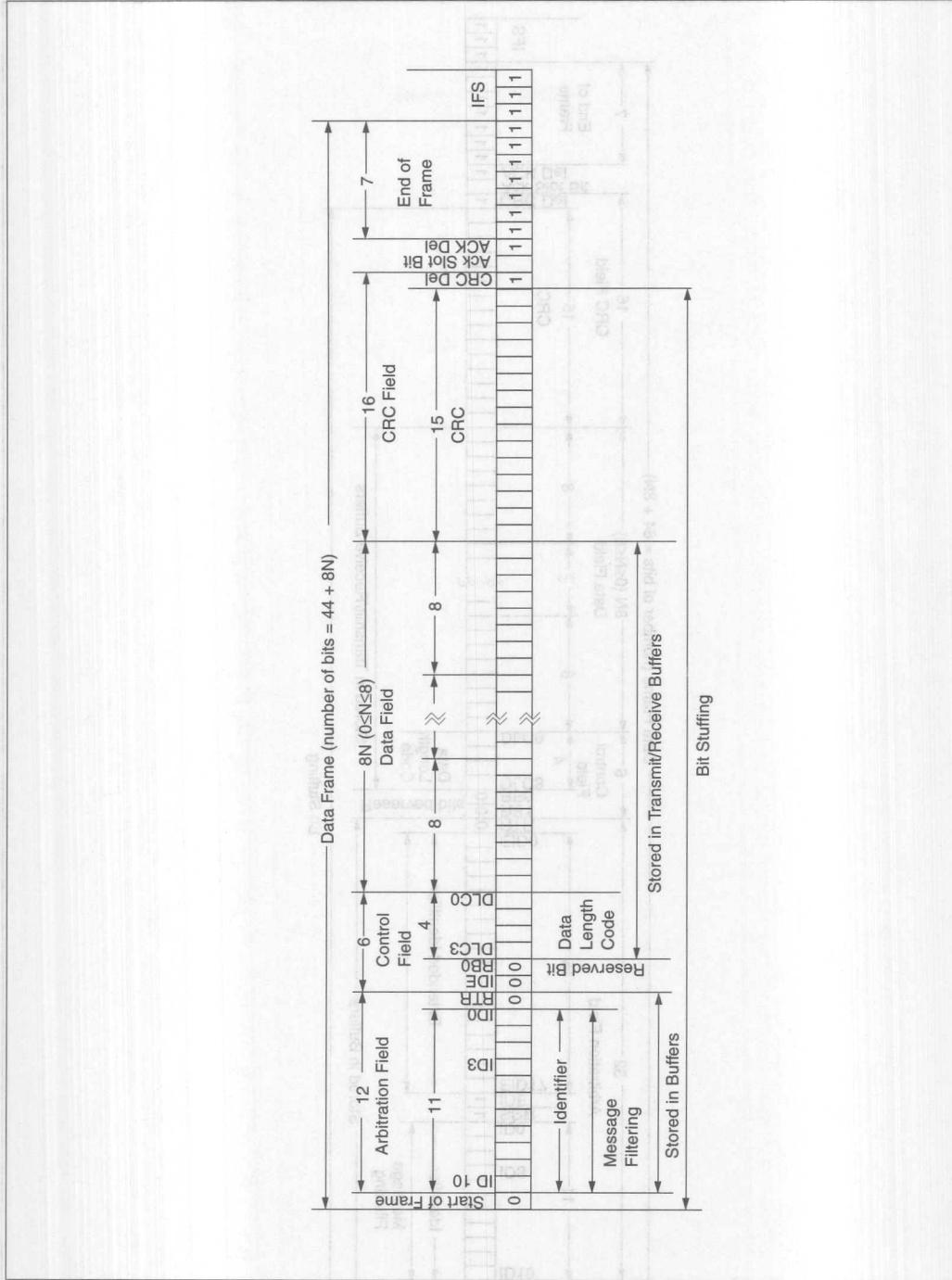


FIGURE 2-1: Standard Data Frame

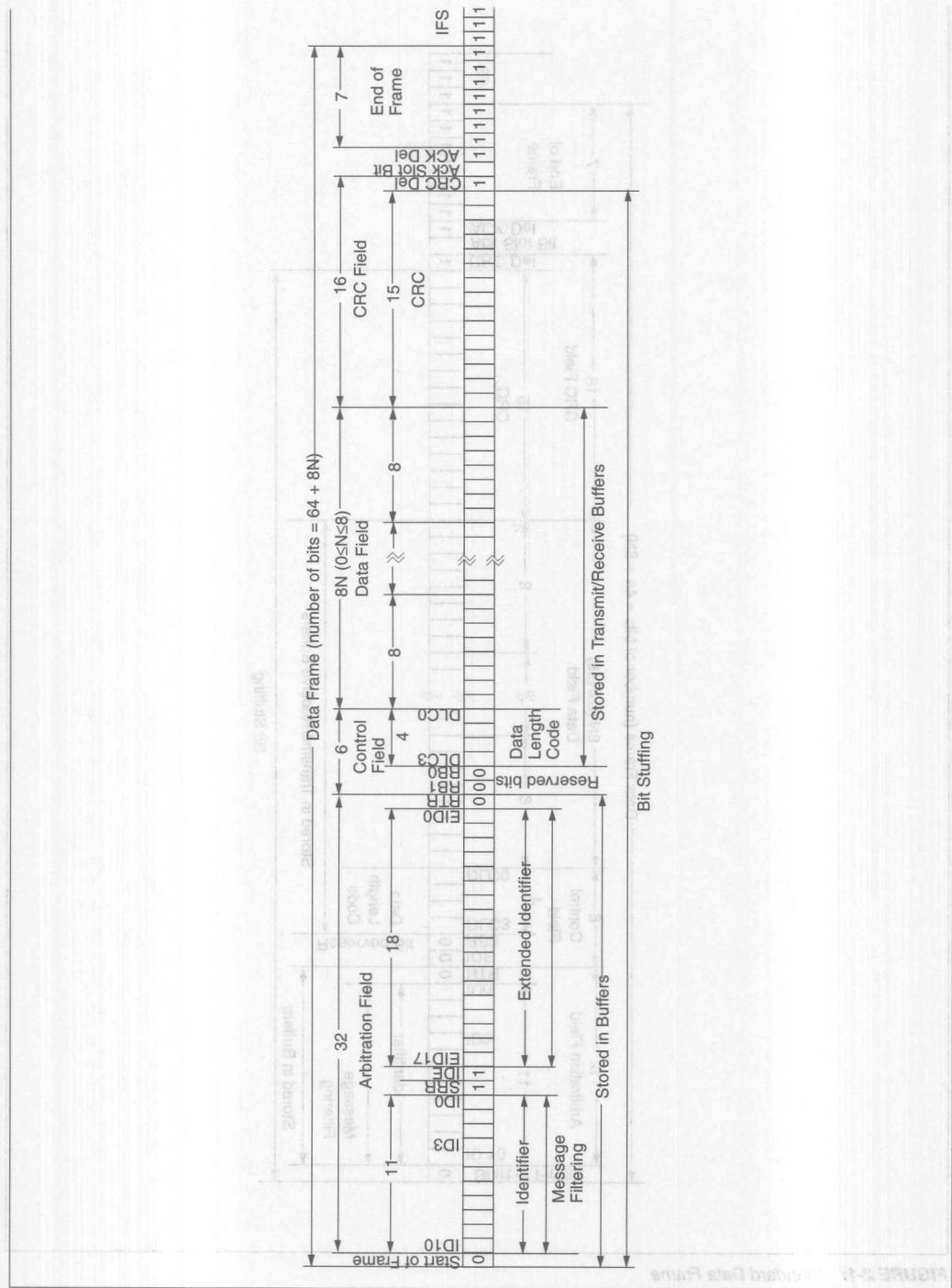
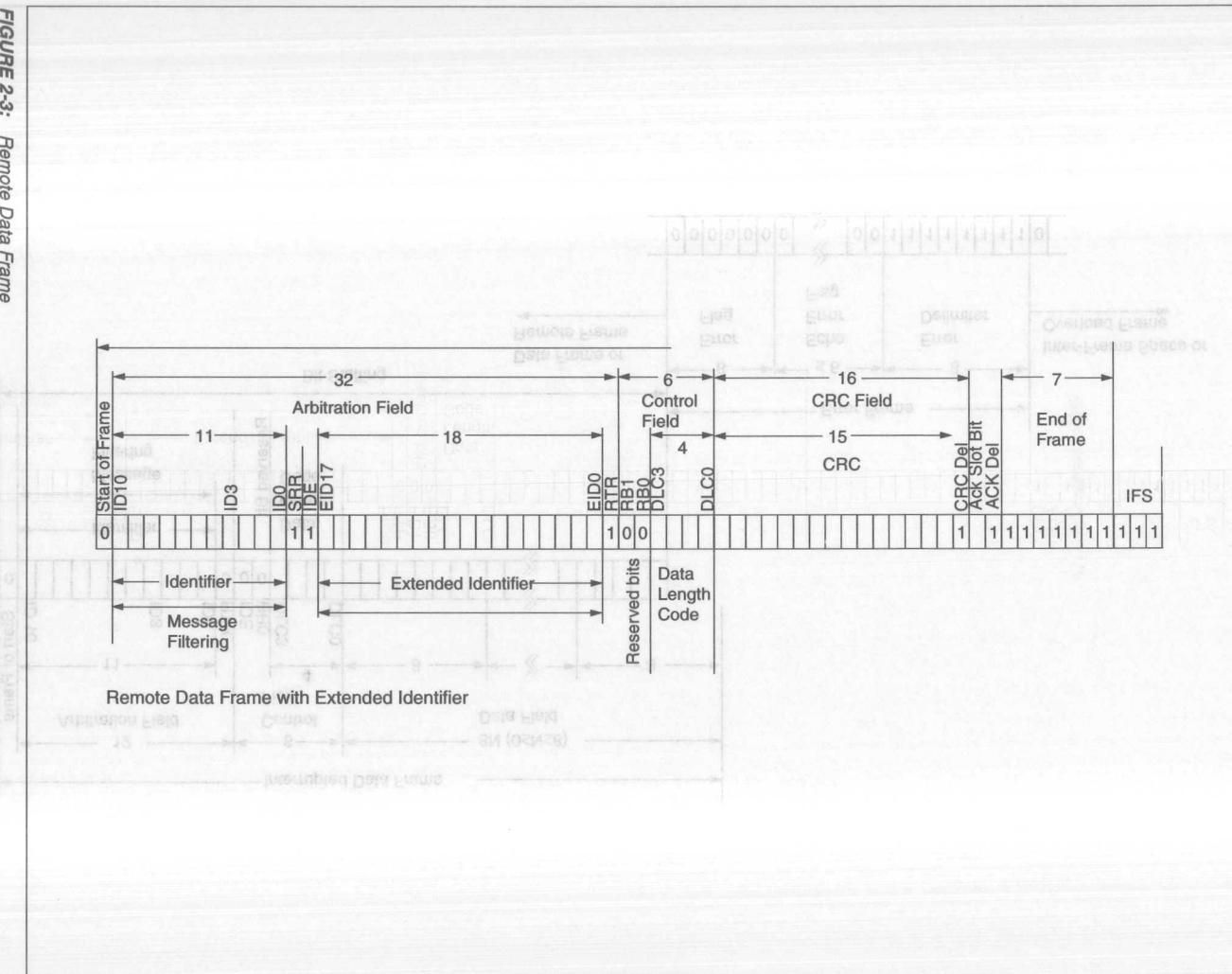


FIGURE 2-2: Extended Data Frame



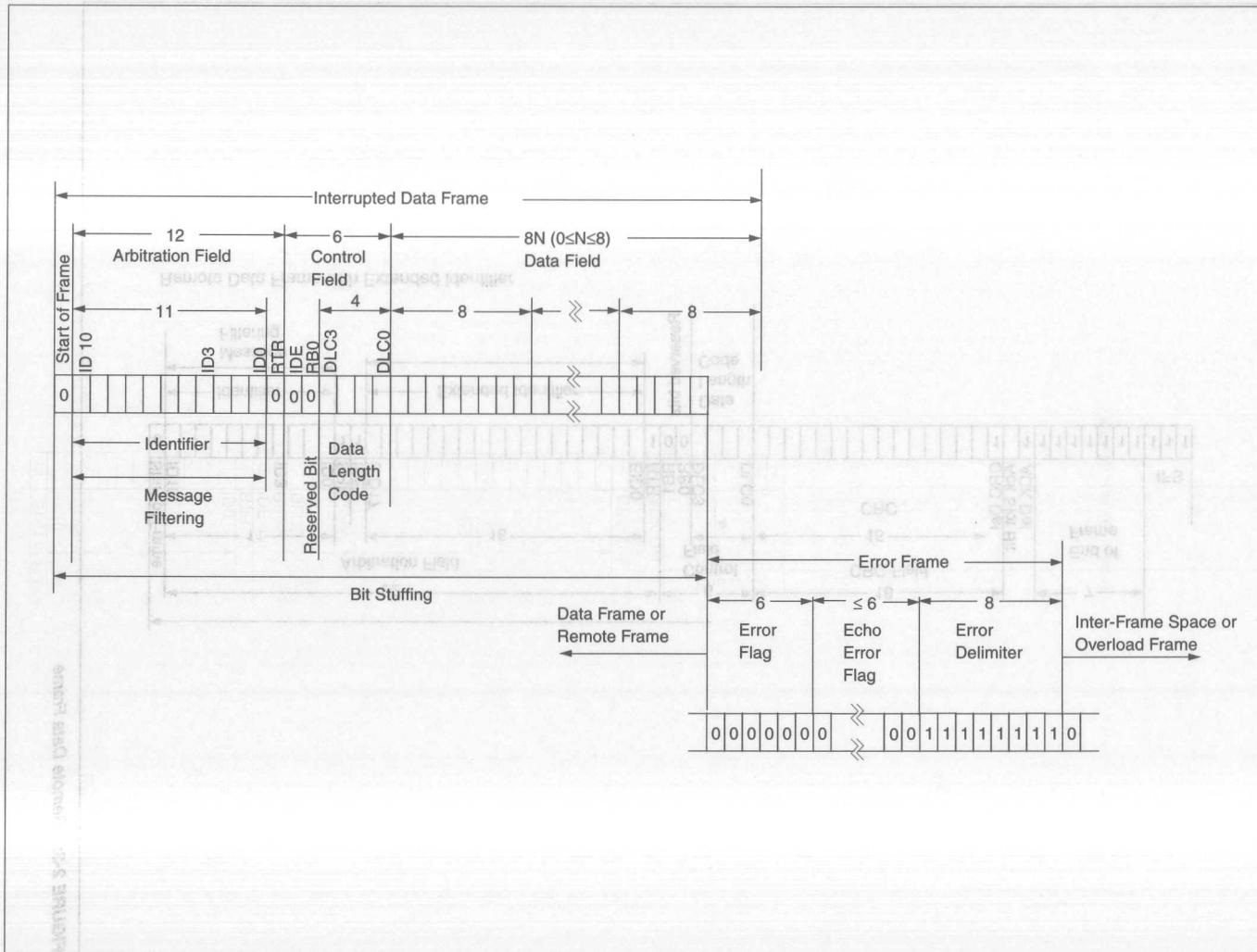


FIGURE 2-4: Error Frame

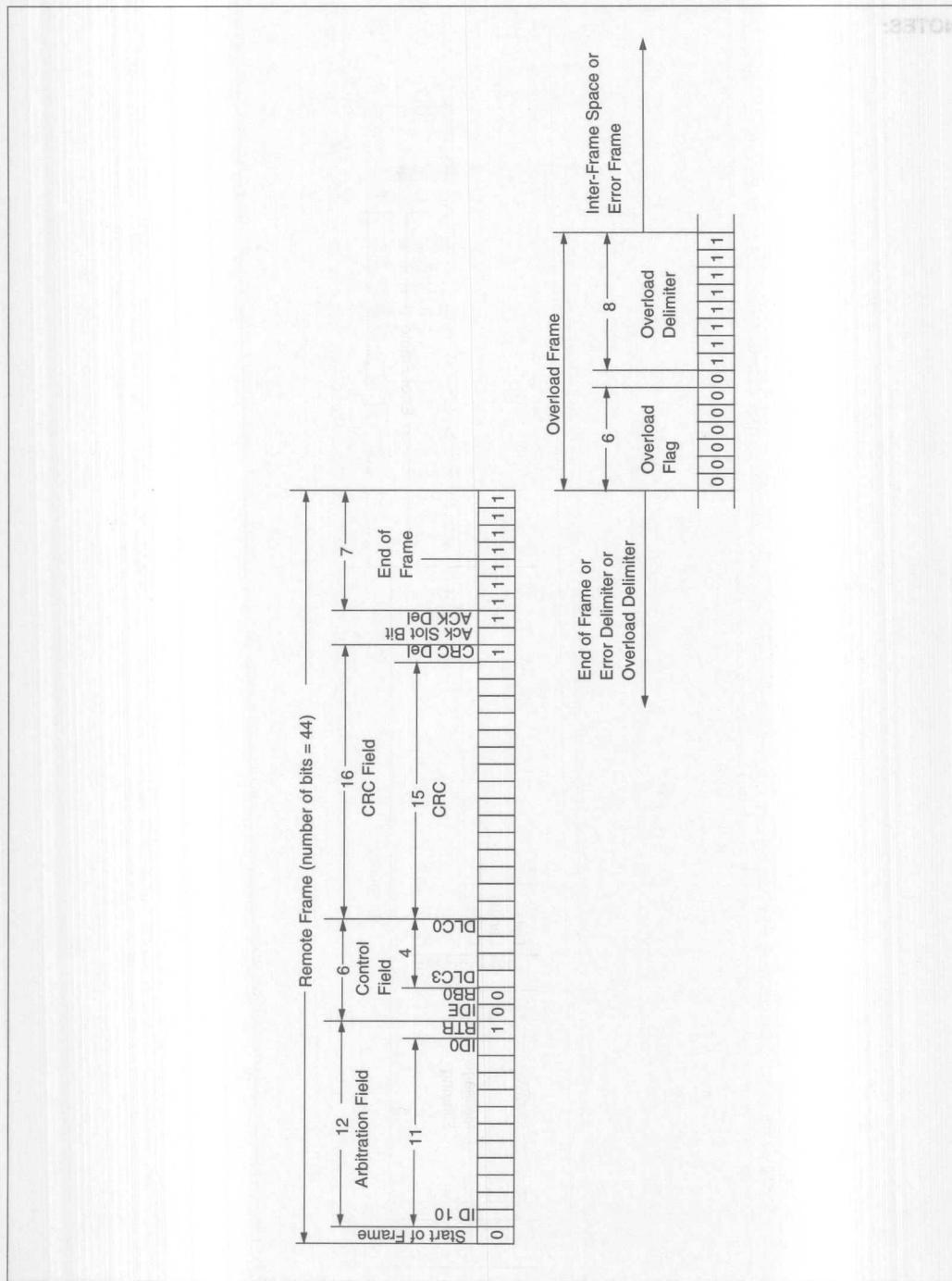
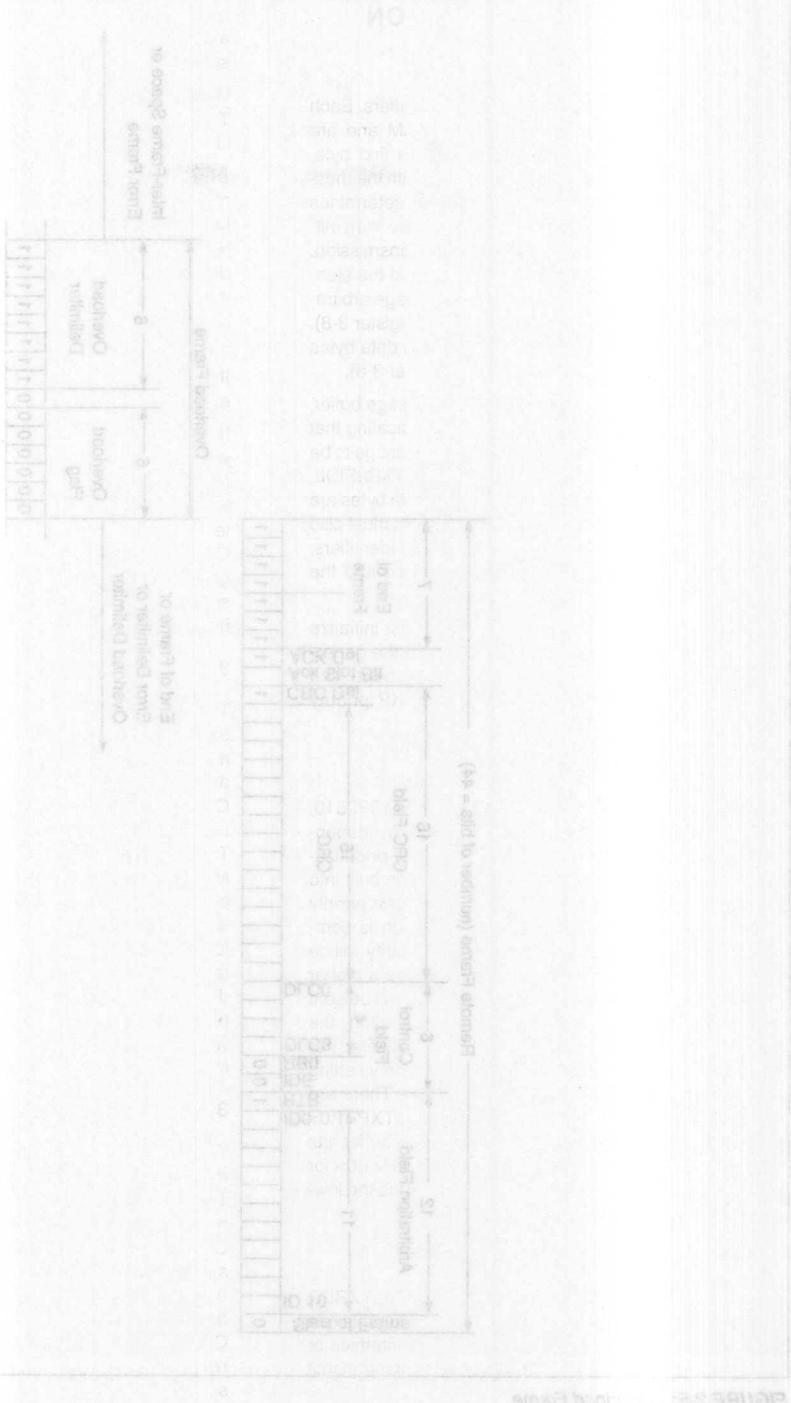


FIGURE 2-5: Overload Frame

MCP2510

NOTES:



3.0 MESSAGE TRANSMISSION

3.1 Transmit Buffers

The MCP2510 implements three Transmit Buffers. Each of these buffers occupies 14 bytes of SRAM and are mapped into the device memory maps. The first byte, TXBNCTRL, is a control register associated with the message buffer. The information in this register determines the conditions under which the message will be transmitted and indicates the status of the message transmission. (see Register 3-2). Five bytes are used to hold the standard and extended identifiers and other message arbitration information (see Register 3-3 through Register 3-8). The last eight bytes are for the eight possible data bytes of the message to be transmitted (see Register 3-8).

For the MCU to have write access to the message buffer, the TXBNCTRL.TXREQ bit must be clear, indicating that the message buffer is clear of any pending message to be transmitted. At a minimum, the TXBNSIDH, TXBNSIDL, and TXBNDLC registers must be loaded. If data bytes are present in the message, the TXBNDr registers must also be loaded. If the message is to use extended identifiers, the TXBNEIDm registers must also be loaded and the TXBNSIDL.EXIDE bit set.

Prior to sending the message, the MCU must initialize the CANINTE.TXINE bit to enable or disable the generation of an interrupt when the message is sent. The MCU must also initialize the TXBNCTRL.TXP priority bits (see Section 3.2).

3.2 Transmit Priority

Transmit priority is a prioritization, within the MCP2510, of the pending transmittable messages. This is independent from, and not necessarily related to, any prioritization implicit in the message arbitration scheme built into the CAN protocol. Prior to sending the SOF, the priority of all buffers that are queued for transmission is compared. The transmit buffer with the highest priority will be sent first. For example, if transmit buffer 0 has a higher priority setting than transmit buffer 1, buffer 0 will be sent first. If two buffers have the same priority setting, the buffer with the highest buffer number will be sent first. For example, if transmit buffer 1 has the same priority setting as transmit buffer 0, buffer 1 will be sent first. There are four levels of transmit priority. If TXBNCTRL.TXP<1:0> for a particular message buffer is set to 11, that buffer has the highest possible priority. If TXBNCTRL.TXP<1:0> for a particular message buffer is 00, that buffer has the lowest possible priority.

3.3 Initiating Transmission

To initiate message transmission the TXBNCTRL.TXREQ bit must be set for each buffer to be transmitted. This can be done by writing to the register via the SPI interface or by setting the TXNRTS pin low for the particular transmit

buffer(s) that are to be transmitted. If transmission is initiated via the SPI interface, the TXREQ bit can be set at the same time as the TXP priority bits.

When TXBNCTRL.TXREQ is set, the TXBNCTRL.ABTF, TXBNCTRL.MLOA and TXBNCTRL.TXERR bits will be cleared.

Setting the TXBNCTRL.TXREQ bit does not initiate a message transmission, it merely flags a message buffer as ready for transmission. Transmission will start when the device detects that the bus is available. The device will then begin transmission of the highest priority message that is ready.

When the transmission has completed successfully the TXBNCTRL.TXREQ bit will be cleared, the CANINTF.TXNIF bit will be set, and an interrupt will be generated if the CANINTE.TXNIE bit is set.

If the message transmission fails, the TXBNCTRL.TXREQ will remain set indicating that the message is still pending for transmission and one of the following condition flags will be set. If the message started to transmit but encountered an error condition, the TXBNCTRL.TXERR and the CANINTF.MERRF bits will be set and an interrupt will be generated on the INT pin if the CANINTE.MERRE bit is set. If the message lost arbitration the TXBNCTRL.MLOA bit will be set.

3.4 TXnRTS Pins

The TXnRTS Pins are input pins that can be configured as request-to-send inputs, which provides a secondary means of initiating the transmission of a message from any of the transmit buffers, or as standard digital inputs. Configuration and control of these pins is accomplished using the TXRTSCTRL register (see Register 3-2). The TXRTSCTRL register can only be modified when the MCP2510 is in configuration mode (see Section 9.0). If configured to operate as a request to send pin, the pin is mapped into the respective TXBNCTRL.TXREQ bit for the transmit buffer. The TXREQ bit is latched by the falling edge of the TXNRTS pin. The TXNRTS pins are designed to allow them to be tied directly to the RXNBF pins to automatically initiate a message transmission when the RXNBF pin goes low. The TXNRTS pins have internal pullup resistors of 100K ohms (nominal).

3.5 Aborting Transmission

The MCU can request to abort a message in a specific message buffer by clearing the associated TXBNCTRL.TXREQ bit. Also, all pending messages can be requested to be aborted by setting the CANCTRL.ABAT bit. If the CANCTRL.ABAT bit is set to abort all pending messages, the user MUST reset this bit (typically after the user verifies that all TXREQ bits have been cleared) to continue transmit messages. The CANCTRL.ABTF flag will only be set if the abort was requested via the CANCTRL.ABAT bit. Aborting a message by resetting the TXREQ bit does NOT cause the ABTF bit to be set.

Only messages that have not already begun to be transmitted can be aborted. Once a message has begun transmission, it will not be possible for the user to reset the TXBNCTRL.TXREQ bit. After transmission

of a message has begun, if an error occurs on the bus or if the message loses arbitration, the message will be retransmitted regardless of a request to abort.

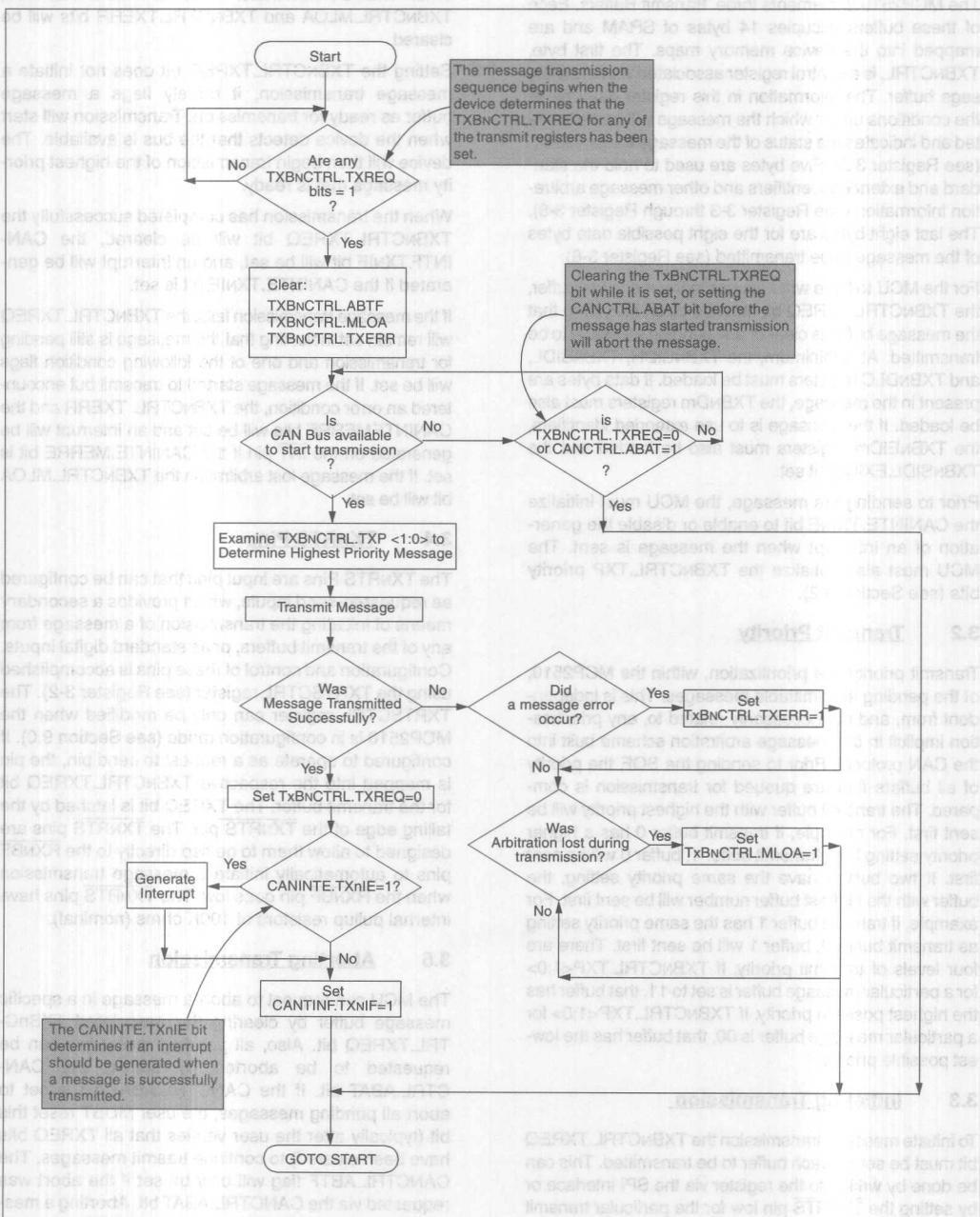


FIGURE 3-1: Transmit Message Flowchart

U-0	R-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0	x-0	x-0	0-U	0-U
—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	bit 7	bit 0	R = Readable bit	W = Writable bit
										C = Bit can be cleared by MCU but not set	U = Unimplemented - reads as '0'

bit 7: **Unimplemented:** Reads as '0'

bit 6: **ABTF:** Message Aborted Flag
1 = Message was aborted
0 = Message completed transmission successfully

bit 5: **MLOA:** Message Lost Arbitration
1 = Message lost arbitration while being sent
0 = Message did not lose arbitration while being sent

bit 4: **TXERR:** Transmission Error Detected
1 = A bus error occurred while the message was being transmitted
0 = No bus error occurred while the message was being transmitted

bit 3: **TXREQ:** Message Transmit Request
1 = Buffer is currently pending transmission
(MCU sets this bit to request message be transmitted - bit is automatically cleared when the message is sent)
0 = Buffer is not currently pending transmission
(MCU can clear this bit to request a message abort)

bit 2: **Unimplemented:** Reads as '0'

bit 1: **TXP<1:0>:** Transmit Buffer Priority
11 = Highest Message Priority
10 = High Intermediate Message Priority
01 = Low Intermediate Message Priority
00 = Lowest Message Priority

REGISTER 3-1: TXBNCTRL Transmit Buffer N Control Register (ADDRESS: 30h, 40h, 50h)

SIID0	SIID1	SIID2	SIID3	SIID4	SIID5	SIID6	SIID7	SIID8	SIID9	SIID10	SIID11
RI sidesetR = R RI sidesetW = W RI beneathR = O RI beneathW = O RI belowR = U RI belowW = U RI aboveR = U RI aboveW = U	RI	RI									

REGISTER 3-2: TXRTSCTRL - TXNRTS Pin Control and Status Register (ADDRESS: 0Dh)

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SID10 | SID9 | SID8 | SID7 | SID6 | SID5 | SID4 | SID3 |
| bit 7 | | | | bit 0 | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |
|
 | | | | | | | |

REGISTER 3-3: TXBNSIDH - Transmit Buffer N Standard Identifier High (ADDRESS: 31h, 41h, 51h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7				bit 0			
bit 7-5: SID<2:0>: Standard Identifier Bits <2:0>							
bit 4: Unimplemented: Reads as '0'							
bit 3: EXIDE: Extended Identifier Enable							
1 = Message will transmit extended identifier							
0 = Message will transmit standard identifier							
bit 2: Unimplemented: Reads as '0'							
bit 1-0: EID<17:16>: Extended Identifier Bits <17:16>							

REGISTER 3-4: TXBNSIDL - Transmit Buffer N Standard Identifier Low (ADDRESS: 32h, 42h, 52h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7				bit 0			
bit 7-0: EID<15:8>: Extended Identifier Bits <15:8>							

REGISTER 3-5: TXBNEID8 - Transmit Buffer N Extended Identifier High (ADDRESS: 33h, 43h, 53h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7				bit 0			
bit 7-0: EID<7:0>: Extended Identifier Bits <7:0>							

REGISTER 3-6: TXBNEID0 - Transmit Buffer N Extended Identifier LOW (ADDRESS: 34h, 44h, 54h)

R = Readable bit
W = Writable bit
C = Bit can be cleared by MCU but not set
U = Unimplemented - reads as '0'
- n = Value at POR reset

MCP2510

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| — | RTR | — W | — | DLC3 | DLC2 | DLC1 | DLC0 |
| bit 7 | — | — | — | — | — | — | — |

R = Readable bit
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR reset

bit 7: **Unimplemented:** Reads as '0'
 bit 6: **RTR:** Remote Transmission Request Bit
 1 = Transmitted Message will be a Remote Transmit Request
 0 = Transmitted Message will be a Data Frame
 bit 5-4: **Unimplemented:** Reads as '0'
 bit 3-0: **DLC<3:0>:** Data Length Code
 Sets the number of data bytes to be transmitted (0 to 8 bytes)
NOTE: It is possible to set the DLC to a value greater than 8, however only 8 bytes are transmitted

REGISTER 3-7: TXBNDLC - Transmit Buffer N Data Length Code (ADDRESS: 35h, 45h, 55h)

| R/W-x |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TXBNDm7 | TXBNDm6 | TXBNDm5 | TXBNDm4 | TXBNDm3 | TXBNDm2 | TXBNDm1 | TXBNDm0 |
| bit 7 | — | — | — | — | — | — | — |

R = Readable bit
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR reset

bit 7-0: **TXBNDM7:TXBNDM0:** Transmit Buffer N Data Field Byte m

REGISTER 3-8: TXBNDM - Transmit Buffer N Data Field Byte m (ADDRESS: 36h-3Dh, 46h-4Dh, 56h-5Dh)

0	1	2	3	4	5	6	7
E1D0	E1D1	E1D2	E1D3	E1D4	E1D5	E1D6	E1D7
0 fid							

4.0 MESSAGE RECEPTION

4.1 Receive Message Buffering

The MCP2510 includes two full receive buffers with multiple acceptance filters for each. There is also a separate Message Assembly Buffer (MAB) which acts as a third receive buffer (see Figure 4-1).

4.2 Receive Buffers

Of the three Receive Buffers, the MAB is always committed to receiving the next message from the bus. The remaining two receive buffers are called RXB0 and RXB1 and can receive a complete message from the protocol engine. The MCU can access one buffer while the other buffer is available for message reception or holding a previously received message.

The MAB assembles all messages received. These messages will be transferred to the RXBN buffers (See Register 4-4 to Register 4-9) only if the acceptance filter criteria are met.

Note: The entire contents of the MAB is moved into the receive buffer once a message is accepted. This means that regardless of the type of identifier (standard or extended) and the number of data bytes received, the entire receive buffer is overwritten with the MAB contents. Therefore the contents of all registers in the buffer must be assumed to have been modified when any message is received.

When a message is moved into either of the receive buffers the appropriate CANINTF.RXNIF bit is set. This bit must be cleared by the MCU, when it has completed processing the message in the buffer, in order to allow a new message to be received into the buffer. This bit provides a positive lockout to ensure that the MCU has finished with the message before the MCP2510 attempts to load a new message into the receive buffer. If the CANINTF.RXNIE bit is set an interrupt will be generated on the INT pin to indicate that a valid message has been received.

4.3 Receive Priority

RXB0 is the higher priority buffer and has two message acceptance filters associated with it. RXB1 is the lower priority buffer and has four acceptance filters associated with it. The lower number of acceptance filters makes the match on RXB0 more restrictive and implies a higher priority for that buffer. Additionally, the RXB0CTRL register can be configured such that if RXB0 contains a valid message, and another valid message is received, an overflow error will not occur and the new message will be moved into RXB1 regardless of the acceptance criteria of RXB1. There are also two programmable acceptance filter masks available, one for each receive buffer (see Section 4.5).

When a message is received, bits <3:0> of the RXBNCTRL Register will indicate the acceptance filter number that enabled reception, and whether the received message is a remote transfer request.

The RXBNCTRL.RXM bits set special receive modes. Normally, these bits are set to 00 to enable reception of all valid messages as determined by the appropriate acceptance filters. In this case, the determination of whether or not to receive standard or extended messages is determined by the RFXNSIDL.EXIDE bit in the acceptance filter register. If the RXBNCTRL.RXM bits are set to 01 or 10, the receiver will accept only messages with standard or extended identifiers respectively. If an acceptance filter has the RFXNSIDL.EXIDE bit set such that it does not correspond with the RXBNCTRL.RXM mode, that acceptance filter is rendered useless. These two modes of RXBNCTRL.RXM bits can be used in systems where it is known that only standard or extended messages will be on the bus. If the RXBNCTRL.RXM bits are set to 11, the buffer will receive all messages regardless of the values of the acceptance filters. Also, if a message has an error before the end of frame, that portion of the message assembled in the MAB before the error frame will be loaded into the buffer. This mode has some value in debugging a CAN system and would not be used in an actual system environment.

4.4 RX0BF and RX1BF Pins

In addition to the INT pin which provides an interrupt signal to the MCU for many different conditions, the receive buffer full pins (RX0BF and RX1BF) can be used to indicate that a valid message has been loaded into RXB0 or RXB1, respectively.

The RXBNBF full pins can be configured to act as buffer full interrupt pins or as standard digital outputs. Configuration and status of these pins is available via the BFPCTRL register (Register 4-3). When set to operate in interrupt mode (by setting BFPCTRL.BxBFE and BFPCTRL.BxBFM bits to a 1), these pins are active low and are mapped to the CANINTF.RXNIF bit for each receive buffer. When this bit goes high for one of the receive buffers, indicating that a valid message has been loaded into the buffer, the corresponding RXNBF pin will go low. When the CANINTF.RXNIF bit is cleared by the MCU, then the corresponding interrupt pin will go to the logic high state until the next message is loaded into the receive buffer.

When used as digital outputs the BFPCTRL.BxBFM and BFPCTRL.BxBFE bits must be set to a '1' for the associated buffer. In this mode the state of the pin is controlled by the BFPCTRL.BxBFS bits. Writing a '1' to the BxBFS bit will cause a high level to be driven on the associated buffer full pin, and a '0' will cause the pin to drive low. When using the pins in this mode the state of the pin should be modified only by using the Bit Modify SPI command to prevent glitches from occurring on either of the buffer full pins.

MCP2510

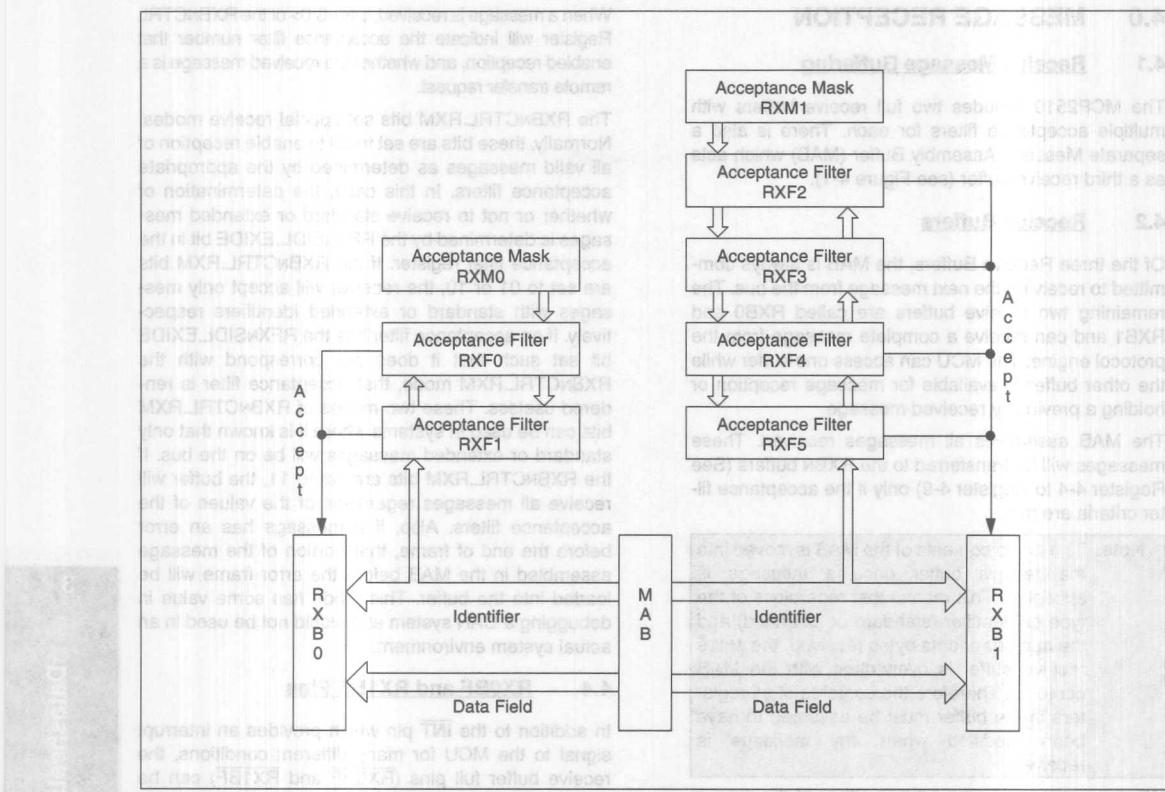


FIGURE 4-1: Receive Buffer Block Diagram

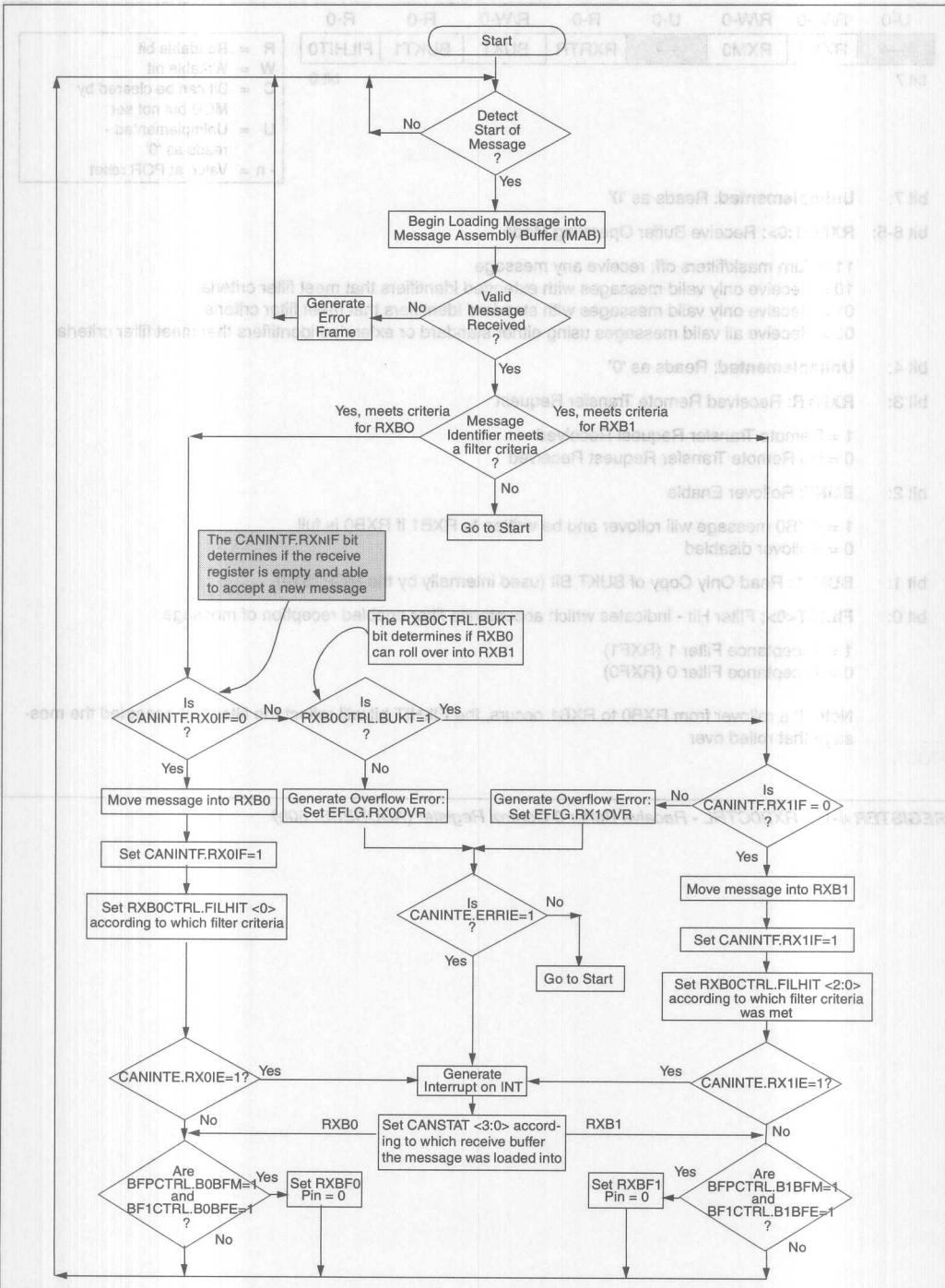


FIGURE 4-2: Message Reception Flowchart

MCP2510

U-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	BUKT	BUKT1	FILHITO

bit 7

bit 0

R = Readable bit
W = Writable bit
C = Bit can be cleared by MCU but not set
U = Unimplemented - reads as '0'
- n = Value at POR reset

bit 7: **Unimplemented:** Reads as '0'

bit 6-5: **RXM<1:0>:** Receive Buffer Operating Mode

- 11 = Turn mask/filters off; receive any message
- 10 = Receive only valid messages with extended identifiers that meet filter criteria
- 01 = Receive only valid messages with standard identifiers that meet filter criteria
- 00 = Receive all valid messages using either standard or extended identifiers that meet filter criteria

bit 4: **Unimplemented:** Reads as '0'

bit 3: **RXRTR:** Received Remote Transfer Request

- 1 = Remote Transfer Request Received
- 0 = No Remote Transfer Request Received

bit 2: **BUKT:** Rollover Enable

- 1 = RXB0 message will rollover and be written to RXB1 if RXB0 is full
- 0 = Rollover disabled

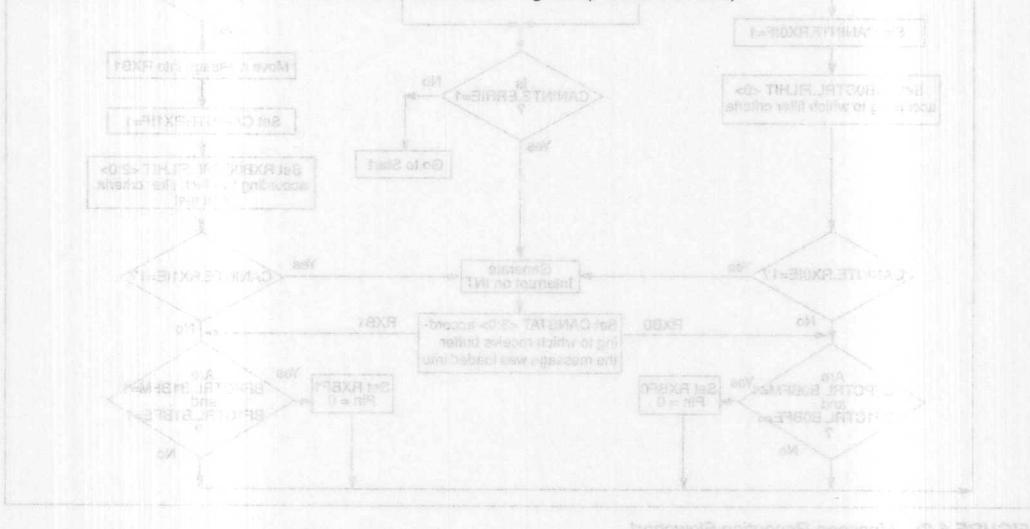
bit 1: **BUKT1:** Read Only Copy of BUKT Bit (used internally by the MCP2510).

bit 0: **FILHIT<0>:** Filter Hit - indicates which acceptance filter enabled reception of message

- 1 = Acceptance Filter 1 (RXF1)
- 0 = Acceptance Filter 0 (RXF0)

Note: If a rollover from RXB0 to RXB1 occurs, the FILHIT bit will reflect the filter that accepted the message that rolled over

REGISTER 4-1: RXB0CTRL - Receive Buffer 0 Control Register (ADDRESS: 60h)



REGISTER 4-2: RXB1CTRL - Receive Buffer 1 Control Register (ADDRESS: 70h)

MCP2510

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0						
—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM					
bit 7								bit 0				
bit 7:	Unimplemented: Reads as '0'							'0' as above				
bit 6:	Unimplemented: Reads as '0'							'0' as above				
bit 5:	B1BFS: RX1BF Pin State (digital output mode only)							'0' as above				
bit 4:	B0BFS: RX0BF Pin State (digital output mode only)							'0' as above				
bit 3:	B1BFE: RX1BF Pin Function Enable							'0' as above				
bit 2:	B0BFE: RX0BF Pin Function Enable							'0' as above				
bit 1:	B1BFM: RX1BF Pin Operation Mode							'0' as above				
bit 0:	B0BFM: RX0BF Pin Operation Mode							'0' as above				

REGISTER 4-3: BFPCTRL - RXnBF Pin Control and Status Register (ADDRESS: 0Ch)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7						bit 0	

REGISTER 4-4: RXBNIDH - Receive Buffer N Standard Identifier High (ADDRESS: 61h, 71h)

R-x	R-x	R-x	R-x	x-R	U-0	x-R	x-Rx	x-R	x-R	U-0
SID2	SID1	SID0	SRR	IDE	—	EID17	EID16	—	—	—
bit 7	—	—	—	—	—	—	—	bit 0	—	—

bit 7-5: **SID<2:0>**: Standard Identifier Bits <2:0>
These bits contain the three least significant bits of the Standard Identifier for the received message

bit 4: **SRR**: Standard Frame Remote Transmit Request Bit (valid only if IDE bit = '0')
1 = Standard Frame Remote Transmit Request Received
0 = Standard Data Frame Received

bit 3: **IDE**: Extended Identifier Flag
This bit indicates whether the received message was a Standard or an Extended Frame
1 = Received message was an Extended Frame
0 = Received message was a Standard Frame

bit 2: **Unimplemented**: Reads as '0'

bit 1-0: **EID<17:16>**: Extended Identifier Bits <17:16>
These bits contain the two most significant bits of the Extended Identifier for the received message

REGISTER 4-5: RXBNSIDL - Receive Buffer N Standard Identifier Low (ADDRESS: 62h, 72h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7	—	—	—	—	—	—	bit 0

bit 7-0: **EID<15:8>**: Extended Identifier Bits <15:8>
These bits hold bits 15 through 8 of the Extended Identifier for the received message

REGISTER 4-6: RXBNEID8 - Receive Buffer N Extended Identifier Mid (ADDRESS: 63h, 73h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7	—	—	—	—	—	—	bit 0

bit 7-0: **EID<7:0>**: Extended Identifier Bits <7:0>
These bits hold the least significant eight bits of the Extended Identifier for the received message

REGISTER 4-7: RXBNEID0 - Receive Buffer N Extended Identifier Low (ADDRESS: 64h, 74h)

REGISTER 4-8: RXBNDLC - Receive Buffer N Data Length Code (ADDRESS: 65h, 75h)

REGISTER 4-9: RXBND_m - Receive Buffer N Data Field Byte m (ADDRESS: 66h-6Dh, 76h-7Dh)

4.5 Message Acceptance Filters and Masks

The Message Acceptance Filters And Masks are used to determine if a message in the message assembly buffer should be loaded into either of the receive buffers (see Figure 4-3). Once a valid message has been received into the MAB, the identifier fields of the message are compared to the filter values. If there is a match, that message will be loaded into the appropriate receive buffer. The filter masks (see Register 4-10 through Register 4-17) are used to determine which bits in the identifier are examined with the filters. A truth table is shown below in Table 4-10 that indicates how each bit in the identifier is compared to the masks and filters to determine if a the message should be loaded into a receive buffer. The mask essentially determines which bits to apply the acceptance filters to. If any mask bit is set to a zero, then that bit will automatically be accepted regardless of the filter bit.

Mask Bit n	Filter Bit n	Message Identifier bit n001	Accept or reject bit n
0	X	W = X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

Note: X = don't care

TABLE 4-10: Filter/Mask Truth Table

As shown in the Receive Buffers Block Diagram (Figure 4-1), acceptance filters RXF0 and RXF1, and filter mask RXM0 are associated with RXB0. Filters RXF2, RXF3, RXF4, and RXF5 and mask RXM1 are associated with RXB1. When a filter matches and a message is loaded into the receive buffer, the filter number that enabled the message reception is loaded into the RXBNCTRL register FILHIT bit(s). For RXB1 the RXB1CTRL register contains the FILHIT<2:0> bits. They are coded as follows:

- 101 = Acceptance Filter 5 (RXF5)
- 100 = Acceptance Filter 4 (RXF4)
- 011 = Acceptance Filter 3 (RXF3)
- 010 = Acceptance Filter 2 (RXF2)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0 (RXF0)

Note: 000 and 001 can only occur if the BUKT bit (see Table 4-1) is set in the RXB0CTRL register allowing RXB0 messages to roll over into RXB1.

RXB0CTRL contains two copies of the BUKT bit and the FILHIT<0> bit.

The coding of the BUKT bit enables these three bits to be used similarly to the RXB1CTRL.FILHIT bits and to distinguish a hit on filter RXF0 and RXF1 in either RXB0 or after a roll over into RXB1.

- 111 = Acceptance Filter 1 (RXF1)
- 110 = Acceptance Filter 0 (RXF0)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0

If the BUKT bit is clear, there are six codes corresponding to the six filters. If the BUKT bit is set, there are six codes corresponding to the six filters plus two additional codes corresponding to RXF0 and RXF1 filters that roll over into RXB1.

If more than one acceptance filter matches, the FILHIT bits will encode the binary value of the lowest numbered filter that matched. In other words, if filter RXF2 and filter RXF4 match, FILHIT will be loaded with the value for RXF2. This essentially prioritizes the acceptance filters with a lower number filter having higher priority. Messages are compared to filters in ascending order of filter number.

The mask and filter registers can only be modified when the MCP2510 is in configuration mode (see Section 9.0).

MCP2510

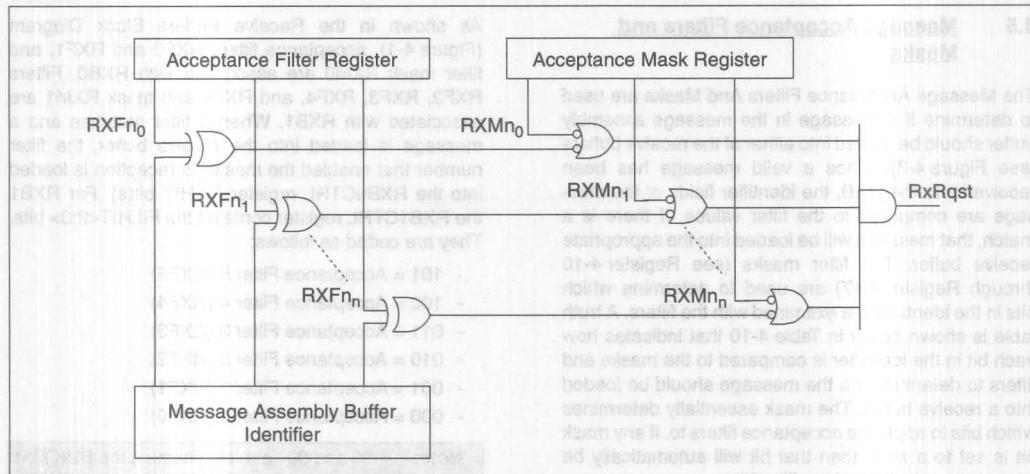


FIGURE 4-3: Message Acceptance Mask and Filter Operation

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7							
bit 6							
bit 5							
bit 4							
bit 3							
bit 2							
bit 1							
bit 0							
bit 7-0: SID<10:3>: Standard Identifier Filter Bits <10:3>							
These bits hold the filter bits to be applied to bits <10:3> of the Standard Identifier portion of a received message							

REGISTER 4-10: RXFNSIDH - Acceptance Filter N Standard Identifier High (Address: 00h, 04h, 08h, 10h, 14h, 18h)

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7						bit 0	
bit 7-5: SID<2:0>: Standard Identifier Filter Bits <2:0> These bits hold the filter bits to be applied to bits <2:0> of the Standard Identifier portion of a received message							
bit 4:	Unimplemented: Reads as '0'						
bit 3:	EXIDE: Extended Identifier Enable 1 = Filter is applied only to Extended Frames 0 = Filter is applied only to Standard Frames						
bit 2:	Unimplemented: Reads as '0'						
bit 1-0:	EID<17:16>: Extended Identifier Filter Bits <17:16> These bits hold the filter bits to be applied to bits <17:16> of the Extended Identifier portion of a received message						

REGISTER 4-11: RXFNSIDL - Acceptance Filter N Standard Identifier Low (Address: 01h, 05h, 09h, 11h, 15h, 19h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7						bit 0	
bit 7-0: EID<15:8>: Extended Identifier Filter Bits <15:8> These bits hold the filter bits to be applied to bits <15:8> of the Extended Identifier portion of a received message							

REGISTER 4-12: RXFNEID8 - Acceptance Filter N Extended Identifier High (Address: 02h, 06h, 0Ah, 12h, 16h, 1Ah)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7						bit 0	
bit 7-0: EID<7:0>: Extended Identifier Filter Bits <7:0> These bits hold the filter bits to be applied to the bits <7:0> of the Extended Identifier portion of a received message							

REGISTER 4-13: RXFNEID0 - Acceptance Filter N Extended Identifier Low (Address: 03h, 07h, 0Bh, 13h, 17h, 1Bh)

R = Readable bit
W = Writable bit
C = Bit can be cleared by MCU but not set
U = Unimplemented - reads as '0'
- n = Value at POR reset

MCP2510

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7							bit 0
bit 7-0: SID<10:3>: Standard Identifier Mask Bits <10:3> These bits hold the mask bits to be applied to bits <10:3> of the Standard Identifier portion of a received message							

R = Readable bit
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR reset

R/W-x	R/W-x	R/W-x	U-0	U-0	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	—	—	EID17	EID16
bit 7							bit 0
bit 7-5: SID<2:0>: Standard Identifier Mask Bits <2:0> These bits hold the mask bits to be applied to bits<2:0> of the Standard Identifier portion of a received message							
bit 4-2: Unimplemented: Reads as '0' bit 1-0: EID<17:16>: Extended Identifier Mask Bits <17:16>							
bit 1-0: EID<17:16>: Extended Identifier Mask Bits <17:16> These bits hold the mask bits to be applied to bits <17:16> of the Extended Identifier portion of a received message							

R = Readable bit = 0
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR,reset

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7							bit 0
bit 7-0: EID<15:8>: Extended Identifier Mask Bits <15:8> These bits hold the mask bits to be applied to bits <15:8> of the Extended Identifier portion of a received message							

R = Readable bit = 0
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR reset

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7							bit 0
bit 7-0: EID<7:0>: Extended Identifier Mask Bits <7:0> These bits hold the mask bits to be applied to bits <7:0> of the Extended Identifier portion of a received message							

R = Readable bit = 0
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR reset

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| EID7 | EID6 | EID5 | EID4 | EID3 | EID2 | EID1 | EID0 |
| bit 7 | | | | | | bit 0 | |
| > | | | | | | | |

bit 7-0: **EID<7:0>**: Extended Identifier Mask Bits <7:0>

These bits hold the mask bits to be applied to bits <7:0> of the Extended Identifier portion of a received message

R = Readable bit
 W = Writable bit
 C = Bit can be cleared by MCU but not set
 U = Unimplemented - reads as '0'
 - n = Value at POR reset

REGISTER 4-17: RXMNEID0 - Acceptance Filter Mask n Extended Identifier Low (Address: 23h-27h)

5.0 BIT TIMING

All nodes on a given CAN bus must have the same nominal bit rate. The CAN protocol uses Non Return to Zero (NRZ) coding which does not encode a clock within the data stream. Therefore, the receive clock must be recovered by the receiving nodes and synchronized to the transmitters clock.

As oscillators and transmission time may vary from node to node, the receiver must have some type of Phase Lock Loop (PLL) synchronized to data transmission edges to synchronize and maintain the receiver clock. Since the data is NRZ coded, it is necessary to include bit stuffing to ensure that an edge occurs at least every six bit times, to maintain the Digital Phase Lock Loop (DPLL) synchronization.

The bit timing of the MCP2510 is implemented using a DPLL that is configured to synchronize to the incoming data, and provide the nominal timing for the transmitted data. The DPLL breaks each bit time into multiple segments made up of minimal periods of time called the time quanta (T_Q).

Bus timing functions executed within the bit time frame, such as synchronization to the local oscillator, network transmission delay compensation, and sample point positioning, are defined by the programmable bit timing logic of the DPLL.

All devices on the CAN bus must use the same bit rate. However, all devices are not required to have the same master oscillator clock frequency. For the different clock frequencies of the individual devices, the bit rate has to be adjusted by appropriately setting the baud rate prescaler and number of time quanta in each segment.

The nominal bit rate is the number of bits transmitted per second assuming an ideal transmitter with an ideal oscillator, in the absence of resynchronization. The nominal bit rate is defined to be a maximum of 1Mb/s.

Nominal Bit Time is defined as:

$$T_{BIT} = 1 / \text{NOMINAL BIT RATE}$$

The nominal bit time can be thought of as being divided into separate non-overlapping time segments. These segments are shown in Figure 5-1.

- Synchronization Segment (Sync_Seg)
- Propagation Time Segment (Prop_Seg)
- Phase Buffer Segment 1 (Phase_Seg1)
- Phase Buffer Segment 2 [Phase_Seg2]

$$\text{Nominal Bit Time} = T_Q * (\text{Sync_Seg} + \text{Prop_Seg} + \text{Phase_Seg1} + \text{Phase_Seg2})$$

The time segments and also the nominal bit time are made up of integer units of time called time quanta or T_Q (see Figure 5-1). By definition, the nominal bit time is programmable from a minimum of 8 T_Q to a maximum of 25 T_Q . Also, by definition the minimum nominal bit time is 1 μ s, corresponding to a maximum 1 Mb/s rate.

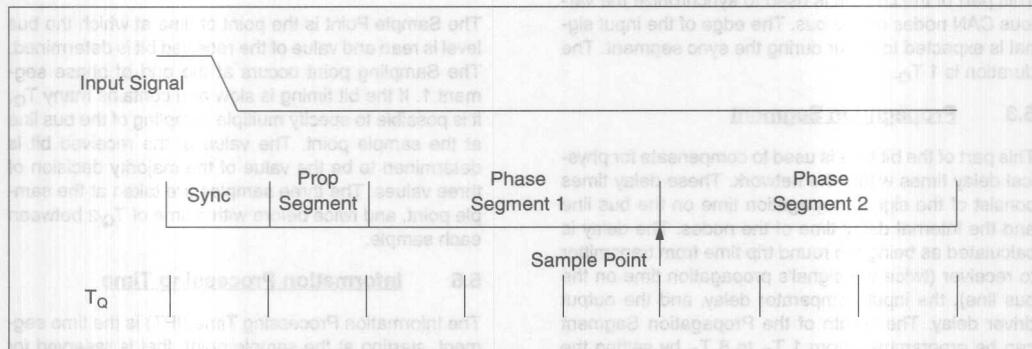


FIGURE 5-1: Bit Time Partitioning

5.1 Time Quanta

The Time Quanta is a fixed unit of time derived from the oscillator period. There is a programmable baud-rate prescaler, with integral values ranging from 1 to 64, in addition to a fixed divide by two for clock generation.

Time quanta is defined as:

$$T_Q = 2 * (\text{Baud Rate} + 1) * T_{OSC}$$

where Baud Rate is the binary value represented by CNF1.BRP<5:0>

For some examples:

If Fosc = 16 MHz, BRP<5:0> = 00h, and Nominal Bit Time = 8 T_Q:

then T_Q = 125 nsec and Nominal Bit Rate = 1 Mb/s

If Fosc = 20 MHz, BRP<5:0> = 01h, and Nominal Bit Time = 8 T_Q:

then T_Q = 200nsec and Nominal Bit Rate = 625 Kb/s

If Fosc = 25 MHz, BRP<5:0> = 3Fh, and Nominal Bit Time = 25 T_Q:

then T_Q = 5.12 usec and Nominal Bit Rate = 7.8 Kb/s

The frequencies of the oscillators in the different nodes must be coordinated in order to provide a system-wide specified nominal bit time. This means that all oscillators must have a T_{OSC} that is a integral divisor of T_Q. It should also be noted that although the number of T_Q is programmable from 4 to 25, the usable minimum is 6 T_Q. Attempting to a bit time of less than 6 T_Q in length is not guaranteed to operate correctly

5.2 Synchronization Segment

This part of the bit time is used to synchronize the various CAN nodes on the bus. The edge of the input signal is expected to occur during the sync segment. The duration is 1 T_Q.

5.3 Propagation Segment

This part of the bit time is used to compensate for physical delay times within the network. These delay times consist of the signal propagation time on the bus line and the internal delay time of the nodes. The delay is calculated as being the round trip time from transmitter to receiver (twice the signal's propagation time on the bus line), the input comparator delay, and the output driver delay. The length of the Propagation Segment can be programmed from 1 T_Q to 8 T_Q by setting the PRSEG2:PRSEG0 bits of the CNF2 register (Table 6-2).

The total delay is calculated from the following individual delays:

1 - 1 * physical bus end to end delay; T_{BUS}

2 - 2 * input comparator delay; T_{COMP} (depends on application circuit)

3 - 2 * output driver delay; T_{DRIVE} (depends on application circuit)

4 - 1 * input to output of CAN controller; T_{CAN} (maximum defined as 1 T_Q + delay ns)

- T_{PROPAGATION} = 2 * (T_{BUS} + T_{COMP} + T_{DRIVE}) + T_{CAN} (maximum bus distance of edge delay of 100m)

- Prop_Seg = T_{PROPAGATION} / T_Q

5.4 Phase Buffer Segments

The Phase Buffer Segments are used to optimally locate the sampling point of the received bit within the nominal bit time. The sampling point occurs between phase segment 1 and phase segment 2. These segments can be lengthened or shortened by the resynchronization process (see Section 5.7.2). Thus, the variation of the values of the phase buffer segments represent the DPLL functionality. The end of phase segment 1 determines the sampling point within a bit time. phase segment 1 is programmable from 1 T_Q to 8 T_Q in duration. Phase segment 2 provides delay before the next transmitted data transition and is also programmable from 1 T_Q to 8 T_Q in duration (however due to IPT requirements the actual minimum length of phase segment 2 is 2 T_Q - see Section 5.6 below), or it may be defined to be equal to the greater of phase segment 1 or the Information Processing Time (IPT). (see Section 5.6).

5.5 Sample Point

The Sample Point is the point of time at which the bus level is read and value of the received bit is determined. The Sampling point occurs at the end of phase segment 1. If the bit timing is slow and contains many T_Q, it is possible to specify multiple sampling of the bus line at the sample point. The value of the received bit is determined to be the value of the majority decision of three values. The three samples are taken at the sample point, and twice before with a time of T_Q/2 between each sample.

5.6 Information Processing Time

The Information Processing Time (IPT) is the time segment, starting at the sample point, that is reserved for calculation of the subsequent bit level. The CAN specification defines this time to be less than or equal to 2 T_Q. The MCP2510 defines this time to be 2 T_Q. Thus, phase segment 2 must be at least 2 T_Q long.

5.7 Synchronization

To compensate for phase shifts between the oscillator frequencies of each of the nodes on the bus, each CAN controller must be able to synchronize to the relevant signal edge of the incoming signal. Synchronization is the process by which the DLL function is implemented. When an edge in the transmitted data is detected, the logic will compare the location of the edge to the expected time (Sync Seg). The circuit will then adjust the values of phase segment 1 and phase segment 2 as necessary. There are two mechanisms used for synchronization.

5.7.1 HARD SYNCHRONIZATION

Hard Synchronization is only done when there is a recessive to dominant edge during a BUS IDLE condition, indicating the start of a message. After hard synchronization, the bit time counters are restarted with Sync Seg. Hard synchronization forces the edge which has occurred to lie within the synchronization segment of the restarted bit time. Due to the rules of synchronization, if a hard synchronization occurs there will not be a resynchronization within that bit time.

5.7.2 RESYNCHRONIZATION

As a result of Resynchronization, phase segment 1 may be lengthened or phase segment 2 may be shortened. The amount of lengthening or shortening of the phase buffer segments has an upper bound given by the Synchronization Jump Width (SJW). The value of the SJW will be added to phase segment 1 (see Figure 5-2) or subtracted from phase segment 2 (see Figure 5-3). The SJW represents the loop filtering of the DLL. The SJW is programmable between 1 T_Q and 4 T_Q .

Clocking information will only be derived from recessive to dominant transitions. The property that only a fixed maximum number of successive bits have the same value ensures resynchronization to the bit stream during a frame.

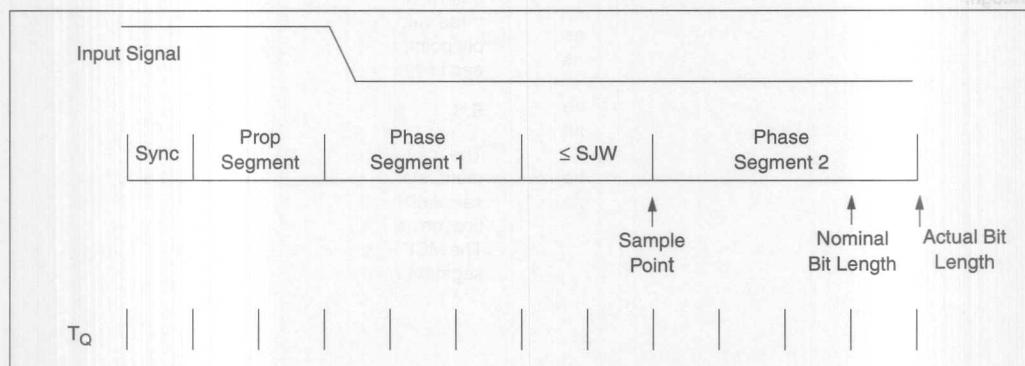


FIGURE 5-2: Lengthening a Bit Period

The phase error of an edge is given by the position of the edge relative to Sync Seg, measured in T_Q . The phase error is defined in magnitude of T_Q as follows:

- $e = 0$ if the edge lies within SYNCSEG.
- $e > 0$ if the edge lies before the SAMPLE POINT.
- $e < 0$ if the edge lies after the SAMPLE POINT of the previous bit.

If the magnitude of the phase error is less than or equal to the programmed value of the synchronization jump width, the effect of a resynchronization is the same as that of a hard synchronization.

If the magnitude of the phase error is larger than the synchronization jump width, and if the phase error is positive, then phase segment 1 is lengthened by an amount equal to the synchronization jump width.

If the magnitude of the phase error is larger than the resynchronization jump width, and if the phase error is negative, then phase segment 2 is shortened by an amount equal to the synchronization jump width.

5.7.3 SYNCHRONIZATION RULES

- Only one synchronization within one bit time is allowed.
- An edge will be used for synchronization only if the value detected at the previous sample point (previously read bus value) differs from the bus value immediately after the edge.
- All other recessive to dominant edges fulfilling rules 1 and 2 will be used for resynchronization with the exception that a node transmitting a dominant bit will not perform a resynchronization as a result of a recessive to dominant edge with a positive phase error.

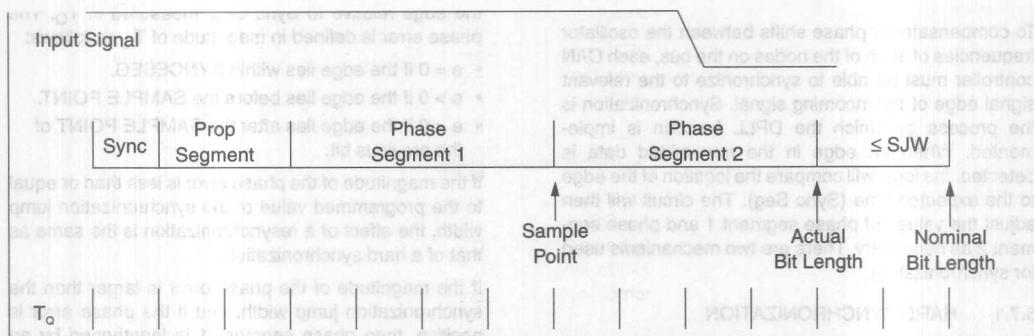


FIGURE 5-3: Shortening a Bit Period

5.8 Programming Time Segments

Some requirements for programming of the time segments:

- Prop Seg + Phase Seg 1 \geq Phase Seg 2
- Prop Seg + Phase Seg 1 $\geq T_{DELAY}$
- Phase Seg 2 $>$ Sync Jump Width

For example, assuming that a 125 kHz CAN baud rate with $F_{OSC} = 20$ MHz is desired:

$T_{osc} = 50$ nsec, choose $BRP<5:0> = 04h$, then $T_Q = 500$ nsec. To obtain 125 kHz, the bit time must be $16 T_Q$.

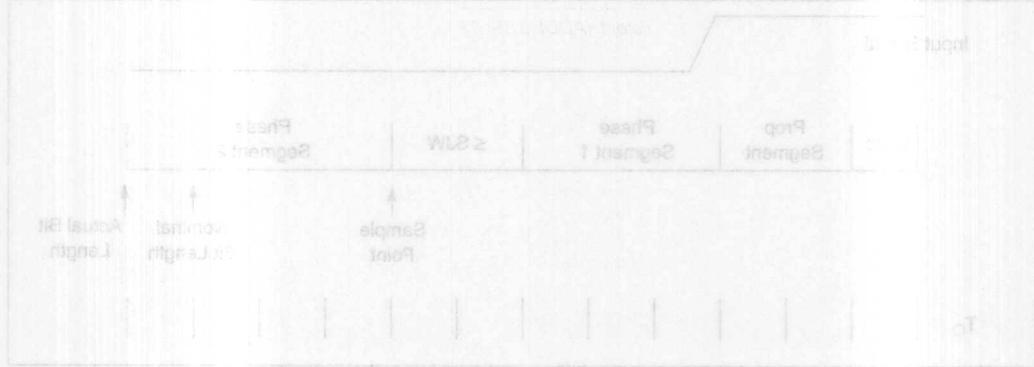
Typically, the sampling of the bit should take place at about 60-70% of the bit time, depending on the system parameters. Also, typically, the T_{DELAY} is $1\cdot2 T_Q$.

Sync Seg = $1 T_Q$; Prop Seg = $2 T_Q$; So setting Phase Seg 1 = $7 T_Q$ would place the sample at $10 T_Q$ after the transition. This would leave $6 T_Q$ for Phase Seg 2.

Since Phase Seg 2 is 6, by the rules, SJW could be the maximum of $4 T_Q$. However, normally a large SJW is only necessary when the clock generation of the different nodes is inaccurate or unstable, such as using ceramic resonators. So an SJW of 1 is typically enough.

5.9 Oscillator Tolerance

The bit timing requirements allow ceramic resonators to be used in applications with transmission rates of up to 125 kbit/sec, as a rule of thumb. For the full bus speed range of the CAN protocol, a quartz oscillator is required. A maximum node-to-node oscillator variation of 1.7% is allowed.



5.10 Bit Timing Configuration Registers

The configuration registers (CNF1, CNF2, CNF3) control the bit timing for the CAN bus interface. These registers can only be modified when the MCP2510 is in configuration mode (see Section 9.0).

5.10.1 CNF1

The BRP<5:0> bits control the baud rate prescaler. These bits set the length of T_Q relative to the OSC1 input frequency, with the minimum length of T_Q being 2 OSC1 clock cycles in length (when BRP<5:0> are set to 000000). The SJW<1:0> bits select the synchronization jump width in terms of number of T_Q 's.

5.10.2 CNF2

The PRSEG<2:0> bits set the length, in T_Q 's, of the propagation segment. The PHSEG1<2:0> bits set the length, in T_Q 's, of phase segment 1. The SAM bit controls how many times the RXCAN pin is sampled. Set-

ting this bit to a '1' causes the bus to be sampled three times; twice at $T_Q/2$ before the sample point, and once at the normal sample point (which is at the end of phase segment 1). The value of the bus is determined to be the value read during at least two of the samples. If the SAM bit is set to a '0' then the RXCAN pin is sampled only once at the sample point. The BTLMODE bit controls how the length of phase segment 2 is determined. If this bit is set to a '1' then the length of phase segment 2 is determined by the PHSEG2<2:0> bits of CNF3 (see Section 5.10.3). If the BTLMODE bit is set to a '0' then the length of phase segment 2 is the greater of phase segment 1 and the information processing time (which is fixed at $2 T_Q$ for the MCP2510).

5.10.3 CNF3

The PHSEG2<2:0> bits set the length, in T_Q's, of Phase Segment 2, if the CNF2.BTLMODE bit is set to a '1'. If the BTLMODE bit is set to a '0' then the PHSEG2<2:0> bits have no effect.

REGISTER 5-1: CNF1 - Configuration Register1 (ADDRESS: 2Ah)

MCP2510

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTLMODE	SAM	PHSEG12	PHSEG11	PHSEG10	PRSEG2	PRSEG1	PRSEG0
bit 7							bit 0
bit 7: BTLMODE: Phase Segment 2 Bit Time Length							
1 = Length of Phase Seg 2 determined by PHSEG22:PHSEG20 bits of CNF3							
0 = Length of Phase Seg 2 is the greater of Phase Seg 1 and IPT (2TQ)							
bit 6: SAM: Sample Point Configuration							
1 = Bus line is sampled three times at the sample point							
0 = Bus line is sampled once at the sample point							
bit 5-3: PHSEG1<2:0>: Phase Segment 1 Length							
111 = Length = 8 x TQ							
000 = Length = 1 x TQ							
bit 2-0 PRSEG<2:0>: Propagation Segment Length							
111 = Length = 8 x TQ							
000 = Length = 1 x TQ							

REGISTER 5-2: CNF2 - Configuration Register2 (ADDRESS: 29h)

U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	WAKFIL	—	—	—	PHSEG22	PHSEG21	PHSEG20
bit 7							bit 0
bit 7: Unimplemented: Reads as '0'							
bit 6: WAKFIL:							
0 = Wake-up filter disabled							
1 = Wake-up filter enabled							
bit 5-3: Unimplemented: Reads as '0'							
bit 2-0 PHSEG2<2:0>: Phase Segment 2 Length							
111 = Length = 8 x TQ							
—							
—							
—							
bit 000 = Length = 1 x TQ							
Note: Minimum valid setting for Phase Segment 2 is 2TQ							

REGISTER 5-3: CNF3 - Configuration Register 3 (ADDRESS: 28h)

6.0 ERROR DETECTION

The CAN protocol provides sophisticated error detection mechanisms. The following errors can be detected.

6.1 CRC Error

With the Cyclic Redundancy Check (CRC), the transmitter calculates special check bits for the bit sequence from the start of a frame until the end of the data field. This CRC sequence is transmitted in the CRC Field. The receiving node also calculates the CRC sequence using the same formula and performs a comparison to the received sequence. If a mismatch is detected, a CRC error has occurred and an error frame is generated. The message is repeated.

6.2 Acknowledge Error

In the acknowledge field of a message, the transmitter checks if the acknowledge slot (which has sent out as a recessive bit) contains a dominant bit. If not, no other node has received the frame correctly. An acknowledge error has occurred; an error frame is generated; and the message will have to be repeated.

6.3 Form Error

If a node detects a dominant bit in one of the four segments including end of frame, interframe space, acknowledge delimiter or CRC delimiter; then a form error has occurred and an error frame is generated. The message is repeated.

6.4 Bit Error

A Bit Error occurs if a transmitter sends a dominant bit and detects a recessive bit or if it sends a recessive bit and detects a dominant bit when monitoring the actual bus level and comparing it to the just transmitted bit. In the case where the transmitter sends a recessive bit and a dominant bit is detected during the arbitration field and the acknowledge slot, no bit error is generated because normal arbitration is occurring.

6.5 Stuff Error

If, between the start of frame and the CRC delimiter, six consecutive bits with the same polarity are detected, the bit stuffing rule has been violated. A stuff error occurs and an error frame is generated. The message is repeated.

6.6 Error States

Detected errors are made public to all other nodes via error frames. The transmission of the erroneous message is aborted and the frame is repeated as soon as possible. Furthermore, each CAN node is in one of the three error states "error-active", "error-passive" or "bus-off" according to the value of the internal error counters. The error-active state is the usual state where the bus node can transmit messages and active error frames (made of dominant bits) without any restrictions. In the error-passive state, messages and passive error frames (made of recessive bits) may be transmitted. The bus-off state makes it temporarily impossible for the station to participate in the bus communication. During this state, messages can neither be received nor transmitted.

6.7 Error Modes and Error Counters

The MCP2510 contains two error counters: the Receive Error Counter (REC) (see Register 6-2), and the Transmit Error Counter (TEC) (see Register 6-1). The values of both counters can be read by the MCU. These counters are incremented or decremented in accordance with the CAN bus specification.

The MCP2510 is error-active if both error counters are below the error-passive limit of 128. It is error-passive if at least one of the error counters equals or exceeds 128. It goes to bus-off if the transmit error counter equals or exceeds the bus-off limit of 256. The device remains in this state, until the bus-off recovery sequence is received. The bus-off recovery sequence consists of 128 occurrences and 11 consecutive recessive bits (see Figure 6-1). Note that the MCP2510, after going bus-off, will recover back to error-active, without any intervention by the MCU, if the bus remains idle for 128 X 11 bit times. If this is not desired, the error interrupt service routine should address this. The current error mode of the MCP2510 can be read by the MCU via the EFLG register (Register 6-3).

Additionally, there is an error state warning flag bit, EFLG:EWARN, which is set if at least one of the error counters equals or exceeds the error warning limit of 96. EWARN is reset if both error counters are less than the error warning limit.

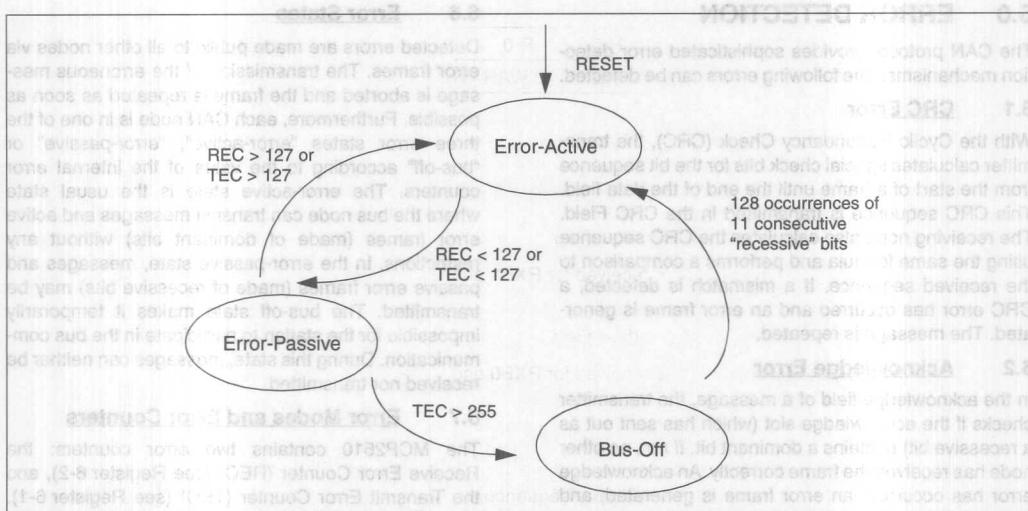
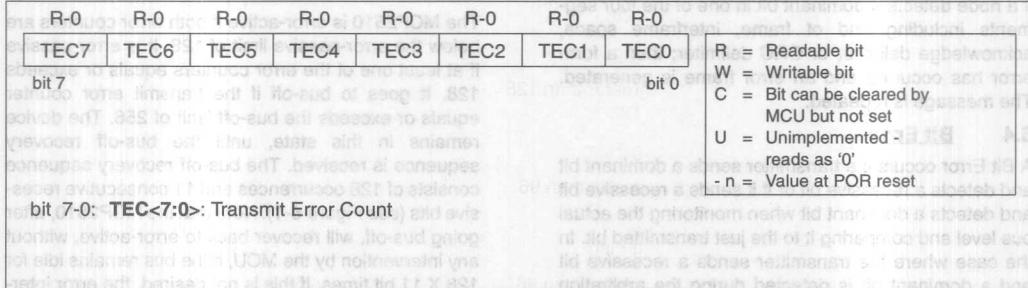
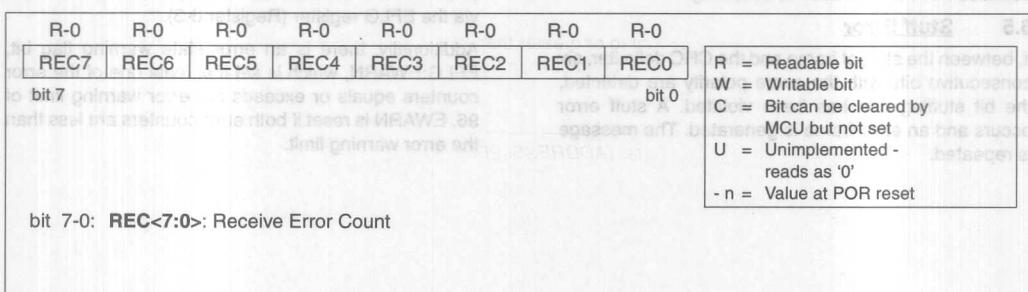


FIGURE 6-1: Error Modes State Diagram



REGISTER 6-1: TEC - Transmitter Error Counter (ADDRESS: 1Ch)



REGISTER 6-2: REC - Receiver Error Counter (ADDRESS: 1Dh)

REGISTER 6-3: EFLG - Error Flag Register (ADDRESS: 2Dh)

NOTES:

7.0 INTERRUPTS

The device has eight sources of interrupts. The CANINTE register contains the individual interrupt enable bits for each interrupt source. The CANINTF register contains the corresponding interrupt flag bit for each interrupt source. When an interrupt occurs the INT pin is driven low by the MCP2510 and will remain low until the interrupt is cleared by the MCU. An interrupt can not be cleared if the respective condition still prevails.

It is recommended that the bit modify command be used to reset flag bits in the CANINTF register rather than normal write operations. This is to prevent unintentionally changing a flag that changes during the write command, potentially causing an interrupt to be missed.

It should be noted that the CANINTF flags are read/write and an interrupt can be generated by the MCU setting any of these bits, provided the associated CANINTE bit is also set.

7.1 Interrupt Code Bits

The source of a pending interrupt is indicated in the CANSTAT.ICOD (interrupt code) bits as indicated in Register 9-2. In the event that multiple interrupts occur, the INT will remain low until all interrupts have been reset by the MCU, and the CANSTAT.ICOD bits will reflect the code for the highest priority interrupt that is currently pending. Interrupts are internally prioritized such that the lower the ICOD value the higher the interrupt priority. Once the highest priority interrupt condition has been cleared, the code for the next highest priority interrupt that is pending (if any) will be reflected by the ICOD bits (see Table 7-1). Note that only those interrupt sources that have their associated CANINTE enable bit set will be reflected in the ICOD bits.

ICOD<2:0>	Boolean Expression
000	ERR•WAK•TX0•TX1•TX2•RX0•RX1
001	ERR
010	ERR•WAK
011	ERR•WAK•TX0
100	ERR•WAK•TX0•TX1
101	ERR•WAK•TX0•TX1•TX2
110	ERR•WAK•TX0•TX1•TX2•RX0•RX0
111	ERR•WAK•TX0•TX1•TX2•RX0•RX1

TABLE 7-1: ICOD<2:0> Decode

7.2 Transmit Interrupt

When the Transmit Interrupt is enabled (CANINTE.TXNIE = 1) an interrupt will be generated on the INT pin when the associated transmit buffer becomes empty and is ready to be loaded with a new message. The CANINTF.TXNIF bit will be set to indicate the source of the interrupt. The interrupt is cleared by the MCU resetting the TXNIF bit to a '0'.

7.3 Receive Interrupt

When the Receive Interrupt is enabled (CANINTE.RXNIE = 1) an interrupt will be generated on the INT pin when a message has been successfully received and loaded into the associated receive buffer. This interrupt is activated immediately after receiving the EOF field. The CANINTF.RXNIF bit will be set to indicate the source of the interrupt. The interrupt is cleared by the MCU resetting the RXNIF bit to a '0'.

7.4 Message Error Interrupt

When an error occurs during transmission or reception of a message the message error flag (CANINTF.MERRF) will be set and, if the CANINTE.MERRE bit is set, an interrupt will be generated on the INT pin. This is intended to be used to facilitate baud rate determination when used in conjunction with listen-only mode.

7.5 Bus Activity Wakeup Interrupt

When the MCP2510 is in sleep mode and the bus activity wakeup interrupt is enabled (CANINTE.WAKIE = 1), an interrupt will be generated on the INT pin, and the CANINTF.WAKIF bit will be set when activity is detected on the CAN bus. This interrupt causes the MCP2510 to exit sleep mode. The interrupt is reset by the MCU clearing the WAKIF bit.

MCP2510

7.6 Error Interrupt

When the error interrupt is enabled (CANINTE.ERRIE = 1) an interrupt is generated on the INT pin if an overflow condition occurs or if the error state of transmitter or receiver has changed. The Error Flag Register (EFLG) will indicate one of the following conditions.

7.6.1 RECEIVER OVERFLOW

An overflow condition occurs when the MAB has assembled a valid received message (the message meets the criteria of the acceptance filters) and the receive buffer associated with the filter is not available for loading of a new message. The associated EFLG.RXNOVR bit will be set to indicate the overflow condition. This bit must be cleared by the MCU.

7.6.2 RECEIVER WARNING

The receive error counter has reached the MCU warning limit of 96.

7.6.3 TRANSMITTER WARNING

The transmit error counter has reached the MCU warning limit of 96.

7.6.4 RECEIVER ERROR-PASSIVE

The receive error counter has exceeded the error-passive limit of 127 and the device has gone to error-passive state.

7.6.5 TRANSMITTER ERROR-PASSIVE

The transmit error counter has exceeded the error-passive limit of 127 and the device has gone to error-passive state.

7.6.6 BUS-OFF

The transmit error counter has exceeded 255 and the device has gone to bus-off state.

7.7 Interrupt Acknowledge

Interrupts are directly associated with one or more status flags in the CANINTF register. Interrupts are pending as long as one of the flags is set. Once an interrupt flag is set by the device, the flag can not be reset by the MCU until the interrupt condition is removed.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	
bit 7:								
bit 6:								
bit 5:								
bit 4:								
bit 3:								
bit 2:								
bit 1:								
bit 0:								

R = Readable bit
W = Writable bit
C = Bit can be cleared by MCU but not set
U = Unimplemented - reads as '0'
-n = Value at POR reset

REGISTER 7-1: CANINTE - Interrupt Enable Register (ADDRESS: 2Bh)

TX2-EXT-TXT-TXT-TXT-AW-	TX1-EXT-TXT-TXT-TXT-AW-	TX0-EXT-TXT-TXT-TXT-AW-	TX1-EXT-TXT-TXT-TXT-AW-	TX0-EXT-TXT-TXT-TXT-AW-	TX1-EXT-TXT-TXT-TXT-AW-	TX0-EXT-TXT-TXT-TXT-AW-	TX1-EXT-TXT-TXT-TXT-AW-	TX0-EXT-TXT-TXT-TXT-AW-
000	000	000	000	000	000	000	000	000
001	001	001	001	001	001	001	001	001
010	010	010	010	010	010	010	010	010
011	011	011	011	011	011	011	011	011
100	100	100	100	100	100	100	100	100
101	101	101	101	101	101	101	101	101
110	110	110	110	110	110	110	110	110
111	111	111	111	111	111	111	111	111

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF
bit 7							bit 0
bit 7: MERRF : Message Error Interrupt Flag							
bit 6: WAKIF : Wakeup Interrupt Flag							
bit 5: ERRIF : Error Interrupt Flag (multiple sources in EFLG register)							
bit 4: TX2IF : Transmit Buffer 2 Empty Interrupt Flag							
bit 3: TX1IF : Transmit Buffer 1 Empty Interrupt Flag							
bit 2: TX0IF : Transmit Buffer 0 Empty Interrupt Flag							
bit 1: RX1IF : Receive Buffer 1 Full Interrupt Flag							
bit 0: RX0IF : Receive Buffer 0 Full Interrupt Flag							
For all bits unless otherwise specified:							
0 = No interrupt pending							
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)							

:250M

bit 7

bit 0

R = Readable bit
W = Writable bit
C = Bit can be cleared by
MCU but not set
U = Unimplemented -
reads as '0'
- n = Value at POR reset

REGISTER 7-2: CANINTF - Interrupt FLAG Register (ADDRESS: 2Ch)

NOTES:

8.0 OSCILLATOR

The MCP2510 is designed to be operated with a crystal or ceramic resonator connected to the OSC1 and OSC2 pins. The MCP2510 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. A typical oscillator circuit is shown in Figure 8-1. The MCP2510 may also be driven by an external clock source connected to the OSC1 pin as shown in Figure 8-2 and Figure 8-3.

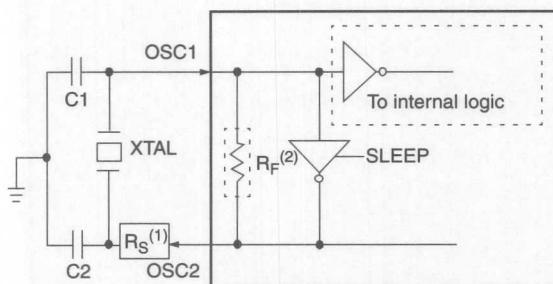
8.1 Oscillator Startup Timer

The MCP2510 utilizes an oscillator startup timer (OST), which holds the MCP2510 in reset, to insure that the oscillator has stabilized before the internal state machine begins to operate. The OST maintains reset for the first 128 OSC1 clock cycles after power up or wake up from sleep mode occurs. It should be noted that no SPI operations should be attempted until after the OST has expired.

8.2 CLKOUT Pin

The clock out pin is provided to the system designer for use as the main system clock or as a clock input for other devices in the system. The CLKOUT has an internal prescaler which can divide F_{osc} by 1, 2, 4 and 8. The CLKOUT function is enabled and the prescaler is selected via the CANCNTRL register (see Register 9-1). The CLKOUT pin will be active upon system reset and default to the slowest speed (divide by 8) so that it can be used as the MCU clock. When sleep mode is requested, the MCP2510 will drive sixteen additional clock cycles on the CLKOUT pin before entering sleep mode. The idle state of the CLKOUT pin in sleep mode is low. When the CLKOUT function is disabled (CANCNTRL.CLKEN = '0') the CLKOUT pin is in a high impedance state.

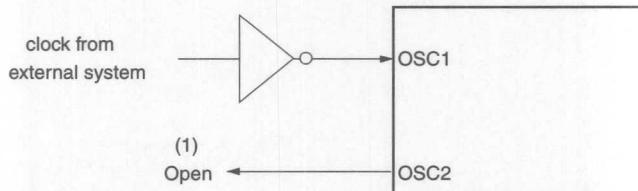
The CLKOUT function is designed to guarantee that $t_{hCLKOUT}$ and $t_{lCLKOUT}$ timings are preserved when the CLKOUT pin function is enabled, disabled, or the prescaler value is changed.



Note 1: A series resistor, R_S , may be required for AT strip cut crystals.

Note 2: The feedback resistor, R_F , is typically in the range of 2 to 10 M Ω .

FIGURE 8-1: Crystal/Ceramic Resonator Operation



Note 1: A resistor to ground may be used to reduce system noise. This may increase system current.

Note 2: Duty cycle restrictions must be observed (see Table 12-2).

FIGURE 8-2: External Clock Source

MCP2510

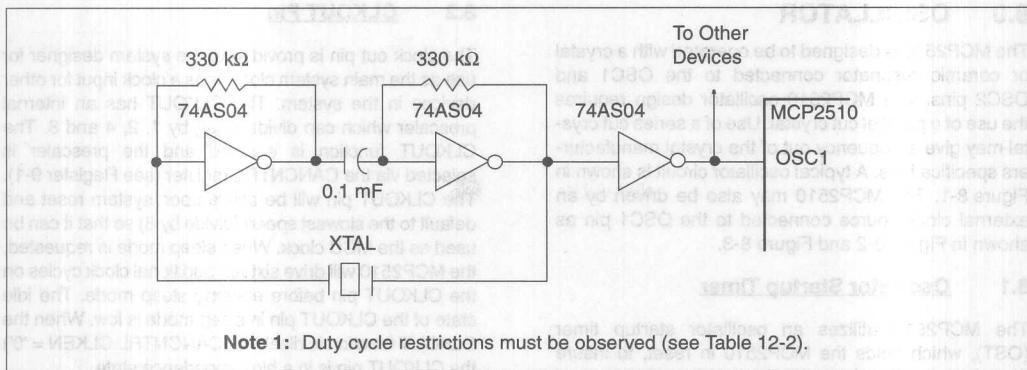


FIGURE 8-3: External Series Resonant Crystal Oscillator Circuit

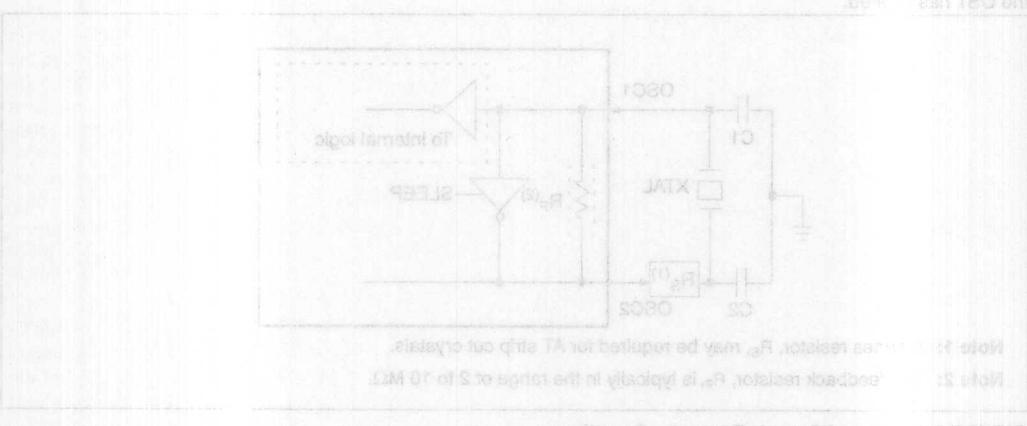


FIGURE 8-4: Detailed External Series Resonant Crystal Oscillator Circuit

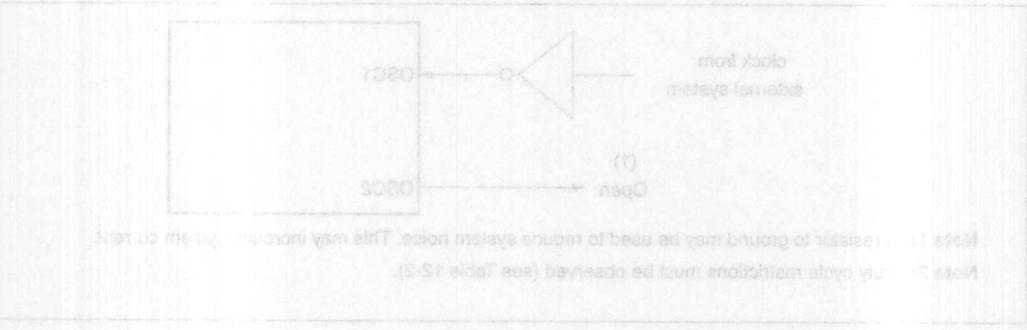


FIGURE 8-5: Detailed External Series Resonant Crystal Oscillator Circuit

9.0 MODES OF OPERATION

The MCP2510 has five modes of operation. These modes are:

1. Configuration Mode.
2. Normal Mode.
3. Sleep Mode.
4. Listen-Only Mode.
5. Loopback Mode.

The operational mode is selected via the CANCTRL.REQOP bits (see Register 9-1). When changing modes, the mode will not actually change until all pending message transmissions are complete. Because of this, the user must verify that the device has actually changed into the requested mode before further operations are executed. Verification of the current operating mode is done by reading the CANSTAT.OPMODE bits (see Register 9-2).

9.1 Configuration Mode

The MCP2510 must be initialized before activation. This is only possible if the device is in the configuration mode. Configuration mode is automatically selected after power-up or a reset, or can be entered from any other mode by setting the CANTRL.REQOP bits to '100'. When configuration mode is entered all error counters are cleared. Configuration mode is the only mode where the following registers are modifiable:

- CNF1, CNF2, CNF3
- TXRTSCTRL
- Acceptance Filter Registers
- Acceptance Mask Registers

Only when the CANSTAT.OPMODE bits read as '100' can the initialization be performed, allowing the configuration registers, acceptance mask registers, and the acceptance filter registers to be written. After the configuration is complete, the device can be activated by programming the CANCTRL.REQOP bits for normal operation mode (or any other mode).

9.2 Sleep Mode

The MCP2510 has an internal sleep mode that is used to minimize the current consumption of the device. The SPI interface remains active even when the MCP2510 is in sleep mode, allowing access to all registers.

To enter sleep mode, the mode request bits are set in the CANCTRL register. The CANSTAT.OPMODE bits indicate whether the device successfully entered sleep mode. These bits should be read after sending the sleep command to the MCP2510. The MCP2510 is active and has not yet entered sleep mode until these bits indicate that sleep mode has been entered. When in internal sleep mode, the wakeup interrupt is still active (if enabled). This is done so the MCU can also be placed into a sleep mode and use the MCP2510 to wake it up upon detecting activity on the bus.

Note: Care must be exercised to not enter sleep mode while the MCP2510 is transmitting a message. If sleep mode is requested while transmitting, the transmission will stop without completing and errors will occur on the bus. Also, the message will remain pending and transmit upon wake up.

When in sleep mode, the MCP2510 stops its internal oscillator. The MCP2510 will wake-up when bus activity occurs or when the MCU sets, via the SPI interface, the CANINTF.WAKIF bit to 'generate' a wake up attempt (the CANINTF.WAKIF bit must also be set in order for the wakeup interrupt to occur). The RXCAN pin will remain in the recessive state while the MCP2510 is in sleep mode. Note that Sleep Mode will be entered immediately, even if a message is currently being transmitted, so it is necessary to insure that all TXREQ bits are clear before setting Sleep Mode.

9.2.1 WAKE-UP FUNCTIONS

The device will monitor the RXCAN pin for activity while it is in sleep mode. If the CANINTE.WAKIE bit is set, the device will wake up and generate an interrupt. Since the internal oscillator is shut down when sleep mode is entered, it will take some amount of time for the oscillator to start up and the device to enable itself to receive messages. The device will ignore the message that caused the wake-up from sleep mode as well as any messages that occur while the device is 'waking up'. The device will wake up in listen-only mode. The MCU must set normal mode before the MCP2510 will be able to communicate on the bus.

The device can be programmed to apply a low-pass filter function to the RXCAN input line while in internal sleep mode. This feature can be used to prevent the device from waking up due to short glitches on the CAN bus lines. The CNF3.WAKFIL bit enables or disables the filter.

9.3 Listen Only Mode

Listen-only mode provides a means for the MCP2510 to receive all messages including messages with errors. This mode can be used for bus monitor applications or for detecting the baud rate in 'hot plugging' situations. For auto-baud detection it is necessary that there are at least two other nodes, which are communicating with each other. The baud rate can be detected empirically by testing different values until valid messages are received. The listen-only mode is a silent mode, meaning no messages will be transmitted while in this state, including error flags or acknowledge signals. The filters and masks can be used to allow only particular messages to be loaded into the receive registers, or the filter masks can be set to all zeros to allow a message with any identifier to pass. The error counters are reset and deactivated in this state. The listen-only mode is activated by setting the mode request bits in the CANCTRL register.

This mode will allow internal transmission of messages from the transmit buffers to the receive buffers without actually transmitting messages on the CAN bus. This mode can be used in system development and testing. In this mode the ACK bit is ignored and the device will allow incoming messages from itself just as if they were coming from another node. The loopback mode is a silent mode, meaning no messages will be transmitted while in this state, including error flags or acknowledge signals. The TXCAN pin will be in a recessive state while the device is in this mode. The filters and masks can be used to allow only particular messages to be loaded into the receive registers. The masks can be set to all zeros to provide a mode that accepts all messages. The loopback mode is activated by setting the mode request bits in the CANCTRL register.

R/W-1	R/W-1	R/W-1	R/W-0	U-0	R/W-1	R/W-1	R/W-1
REQOP2	REQOP1	REQOP0	ABAT	—	CLKEN	CLKPRE1	CLKPRE0

bit 7: **REQOP<2:0>**: Request Operation Mode
000 = Set Normal Operation Mode
001 = Set Sleep Mode
010 = Set Loopback Mode
011 = Set Listen Only Mode
100 = Set Configuration Mode
All other values for REQOP bits are invalid and should not be used

bit 4: **ABAT**: Abort All Pending Transmissions
1 = Request abort of all pending transmit buffers
0 = Terminate request to abort all transmissions

bit 3: **Unimplemented**: Reads as '0'

bit 2: **CLKEN**: CLKOUT Pin Enable
0 = CLKOUT pin disabled (Pin is in high impedance state)
1 = CLKOUT pin enabled

bit 1-0: **CLKPRE<1:0>**: CLKOUT Pin Prescaler
00 = FCLKOUT = System Clock/1
01 = FCLKOUT = System Clock/2
10 = FCLKOUT = System Clock/4
11 = FCLKOUT = System Clock/8

REGISTER 9-1: CANCTRL - CAN Control Register (ADDRESS: xFh)

This is the standard operating mode of the MCP2510. In this mode the device actively monitors all bus messages and generates acknowledge bits, error frames, etc. This is also the only mode in which the MCP2510 will transmit messages over the CAN bus.

R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—

bit 7

bit 0

bit 7-5: OPMOD<2:0>: Operation Mode

000 = Device is in Normal Operation Mode
001 = Device is in Sleep Mode
010 = Device is in Loopback Mode
011 = Device is in Listen Only Mode
100 = Device is in Configuration Mode

bit 4: Unimplemented: Reads as '0'

bit 3-1: ICOD<2:0>: Interrupt Flag Code

000 = No Interrupt
001 = Error Interrupt
010 = Wake Up Interrupt
011 = TXB0 Interrupt
100 = TXB1 Interrupt
101 = TXB2 Interrupt
110 = RXB0 Interrupt
111 = RXB1 Interrupt

bit 0: Unimplemented: Reads as '0'

R = Readable bit
W = Writable bit
C = Bit can be cleared by MCU but not set
U = Unimplemented - reads as '0'
- n = Value at POR reset

REGISTER 9-2: CANSTAT - CAN Status Register (ADDRESS: xEh)

MCP2510

NOTES:

10.0 REGISTER MAP

The register map for the MCP2510 is shown in Table 10-1. Address locations for each register are determined by using the column (higher order 4 bits) and row (lower order 4 bits) values. The registers have been arranged to optimize the sequential reading and

writing of data. Some specific control and status registers allow individual bit modification using the SPI Bit Modify command. The registers that allow this command are shown as shaded locations in Table 10-1. A summary of the MCP2510 control registers is shown in Table 10-2.

Lower Address Bits	Higher Order Address Bits							
	x000 xxxx	x001 xxxx	x010 xxxx	x0011 xxxx	x100 xxxx	x101 xxxx	x110 xxxx	x111 xxxx
0000	RXF0SIDH	RXF3SIDH	RXM0SIDH	TXB0CTRL	TXB1CTRL	TXB2CTRL	RXB0CTRL	RXB1CTRL
0001	RXF0SIDL	RXF3SIDL	RXM0SIDL	TXB0SIDH	TXB1SIDH	TXB2SIDH	RXB0SIDH	RXB1SIDH
0010	RXF0EID8	RXF3EID8	RXM0EID8	TXB0SIDL	TXB1SIDL	TXB2SIDL	RXB0SIDL	RXB1SIDL
0011	RXF0EID0	RXF3EID0	RXM0EID0	TXB0EID8	TXB1EID8	TXB2EID8	RXB0EID8	RXB1EID8
0100	RXF1SIDH	RXF4SIDH	RXM1SIDH	TXB0EID0	TXB1EID0	TXB2EID0	RXB0EID0	RXB1EID0
0101	RXF1SIDL	RXF4SIDL	RXM1SIDL	TXB0DLC	TXB1DLC	TXB2DLC	RXB0DLC	RXB1DLC
0110	RXF1EID8	RXF4EID8	RXM1EID8	TXB0D0	TXB1D0	TXB2D0	RXB0D0	RXB1D0
0111	RXF1EID0	RXF4EID0	RXM1EID0	TXB0D1	TXB1D1	TXB2D1	RXB0D1	RXB1D1
1000	RXF2SIDH	RXF5SIDH	CNF3	TXB0D2	TXB1D2	TXB2D2	RXB0D2	RXB1D2
1001	RXF2SIDL	RXF5SIDL	CNF2	TXB0D3	TXB1D3	TXB2D3	RXB0D3	RXB1D3
1010	RXF2EID8	RXF5EID8	CNF1	TXB0D4	TXB1D4	TXB2D4	RXB0D4	RXB1D4
1011	RXF2EID0	RXF5EID0	CANINTE	TXB0D5	TXB1D5	TXB2D5	RXB0D5	RXB1D5
1100	BFPCTRL	TEC	CANINTF	TXB0D6	TXB1D6	TXB2D6	RXB0D6	RXB1D6
1101	TXRTSCTRL	REC	EFLG	TXB0D7	TXB1D7	TXB2D7	RXB0D7	RXB1D7
1110	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT
1111	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL

Note: Shaded register locations indicate that these allow the user to manipulate individual bits using the 'Bit Modify' Command

TABLE 10-1: CAN Controller Register Map

Register Name	Address (Hex)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/RST Value
BFPCTRL	0C	—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM	--00 0000
TXRTSCTRL	0D	—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM	--xx x000
CANSTAT	xE	OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—	100- 000-
CANCTRL	xF	REQOP2	REQOP1	REQOP0	ABAT	—	CLKEN	CLKPRE1	CLKPRE0	1110 -111
TEC	1C	Transmit Error Counter							0000 0000	
REC	1D	Receive Error Counter							0000 0000	
CNF3	28	—	WAKFIL	—	—	—	PHSEG22	PHSEG21	PHSEG20	-0-- -000
CNF2	29	BTLMODE	SAM	PHSEG12	PHSEG11	PHSEG10	PRSEG2	PRSEG1	PRSEG0	0000 0000
CNF1	2A	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	0000 0000
CANINTE	2B	MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000
CANINTF	2C	MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000
EFLG	2D	RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	0000 0000
TXB0CTRL	30	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXPO	-000 0-00
TXB1CTRL	40	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXPO	-000 0-00
TXB2CTRL	50	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXPO	-000 0-00
RXB0CTRL	60	—	RXM1	RXM0	—	RXRTR	BUKT	BUKT	FILHITO	-00- 0000
RXB1CTRL	70	—	RSM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHITO	-00- 0000

TABLE 10-2: Control Register Summary

MCP2510

NOTES:

11.0 SPI INTERFACE

11.1 Overview

The MCP2510 is designed to interface directly with the Serial Peripheral Interface (SPI) port available on many microcontrollers and supports Mode 0,0 and Mode 1,1. Commands and data are sent to the device via the SI pin, with data being clocked in on the rising edge of SCK. Data is driven out by the MCP2510, on the SO line, on the falling edge of SCK. The CS pin must be held low while any operation is performed. Table 11-1 shows the instruction bytes for all operations. Refer to Figure 11-8 and Figure 11-9 for detailed input and output timing diagrams for both Mode 0,0 and Mode 1,1 operation.

11.2 Read Instruction

The Read Instruction is started by lowering the CS pin. The read instruction is then sent to the MCP2510 followed by the 8-bit address (A7 through A0). After the read instruction and address are sent, the data stored in the register at the selected address will be shifted out on the SO pin. The internal address pointer is automatically incremented to the next address after each byte of data is shifted out. Therefore it is possible to read the next consecutive register address by continuing to provide clock pulses. Any number of consecutive register locations can be read sequentially using this method. The read operation is terminated by raising the CS pin (Figure 11-2).

11.3 Write Instruction

The Write Instruction is started by lowering the CS pin. The write instruction is then sent to the MCP2510 followed by the address and at least one byte of data. It is possible to write to sequential registers by continuing to clock in data bytes, as long as CS is held low. Data will actually be written to the register on the rising edge of the SCK line for the D0 bit. If the CS line is brought high before eight bits are loaded, the write will be aborted for that data byte, previous bytes in the command will have been written. Refer to the timing diagram in Figure 11-3 for more detailed illustration of the byte write sequence.

11.4 Request To Send (RTS) Instruction

The RTS command can be used to initiate message transmission for one or more of the transmit buffers.

The part is selected by lowering the CS pin and the RTS command byte is then sent to the MCP2510. As shown in Figure 11-4, the last 3 bits of this command indicate which transmit buffer(s) are enabled to send. This command will set the TxBnCTRL.TXREQ bit for the respective buffer(s). Any or all of the last three bits can be set in a single command. If the RTS command is sent with nnn=000, the command will be ignored.

11.5 Read Status Instruction

The Read Status Instruction allows single instruction access to some of the often used status bits for message reception and transmission.

The part is selected by lowering the CS pin and the read status command byte, shown in Figure 11-6, is sent to the MCP2510. After the command byte is sent, the MCP2510 will return eight bits of data that contain the status. If additional clocks are sent after the first eight bits are transmitted, the MCP2510 will continue to output the status bits as long as the CS pin is held low and clocks are provided on SCK. Each status bit returned in this command may also be read by using the standard read command with the appropriate register address.

11.6 Bit Modify Instruction

The Bit Modify Instruction provides a means for setting or clearing individual bits in specific status and control registers. This command is not available for all registers. See Section 10.0 (register map) to determine which registers allow the use of this command.

The part is selected by lowering the CS pin and the Bit Modify command byte is then sent to the MCP2510. After the command byte is sent, the address for the register is sent followed by the mask byte and then the data byte. The mask byte determines which bits in the register will be allowed to change. A '1' in the mask byte will allow a bit in the register to change and a '0' will not. The data byte determines what value the modified bits in the register will be changed to. A '1' in the data byte will set the bit and a '0' will clear the bit, provided that the mask for that bit is set to a '1'. (see Figure 11-1)

11.7 Reset Instruction

The Reset Instruction can be used to re-initialize the internal registers of the MCP2510 and set configuration mode. This command provides the same functionality, via the SPI interface, as the RESET pin. The Reset instruction is a single byte instruction which requires selecting the device by pulling CS low, sending the instruction byte, and then raising CS. It is highly recommended that the reset command be sent (or the RESET pin be lowered) as part of the power-on initialization sequence.

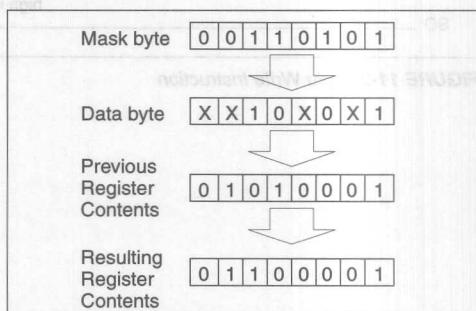


FIGURE 11-1: Bit Modify

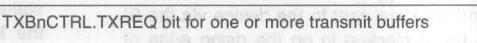
Instruction Name	Instruction Format	Description
RESET	1100 0000	Resets internal registers to default state, set configuration mode
READ	0000 0011	Read data from register beginning at selected address
WRITE	0000 0010	Write data to register beginning at selected address
RTS (Request To Send)	1000 0nnn	Sets TXBnCTRL.TXREQ bit for one or more transmit buffers
		
Read Status	1010 0000	Polling command that outputs status bits for transmit/receive functions
Bit Modify	0000 0101	Bit modify selected registers

TABLE 11-1: SPI Instruction Set

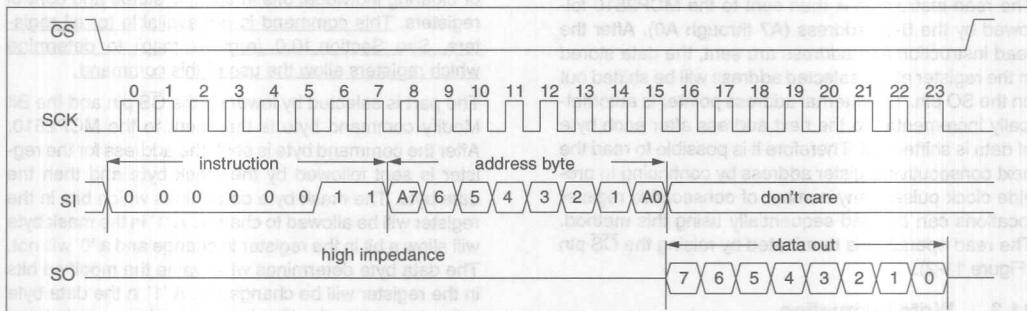


FIGURE 11-2: Read Instruction

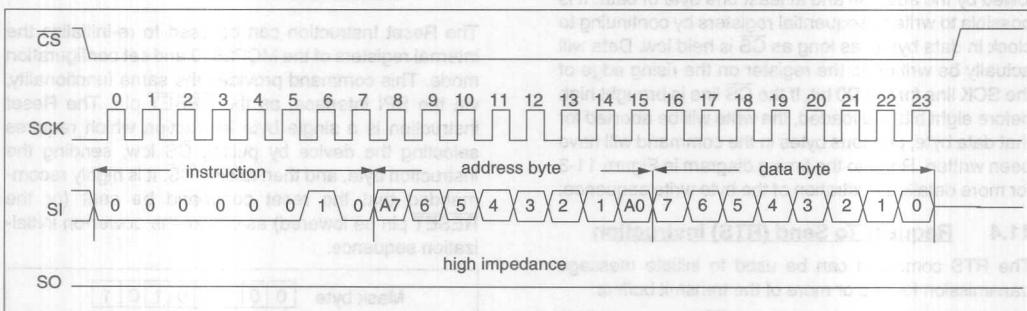


FIGURE 11-3: Byte Write Instruction

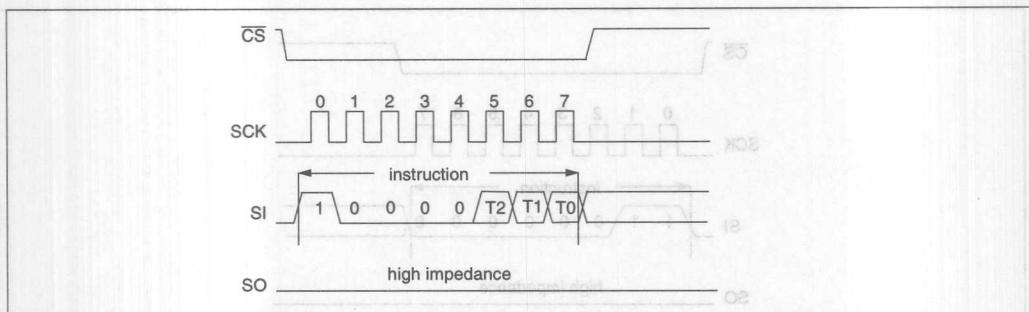


FIGURE 11-4: Request To Send Instruction

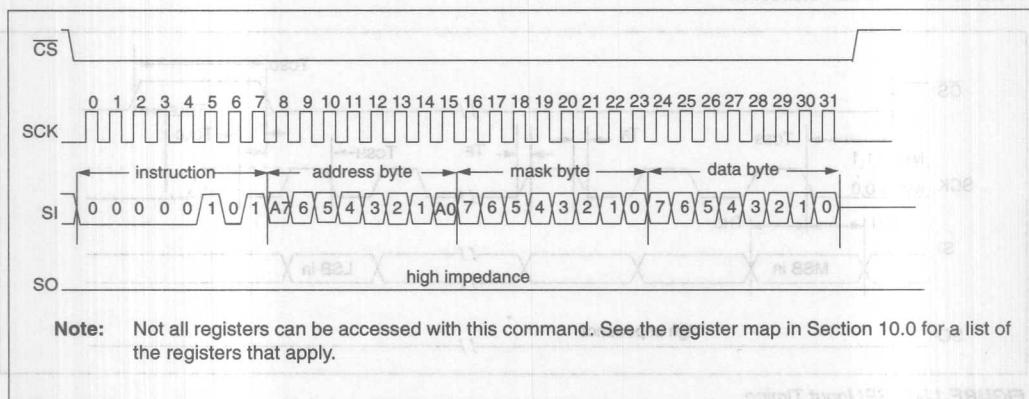


FIGURE 11-5: BIT Modify instruction

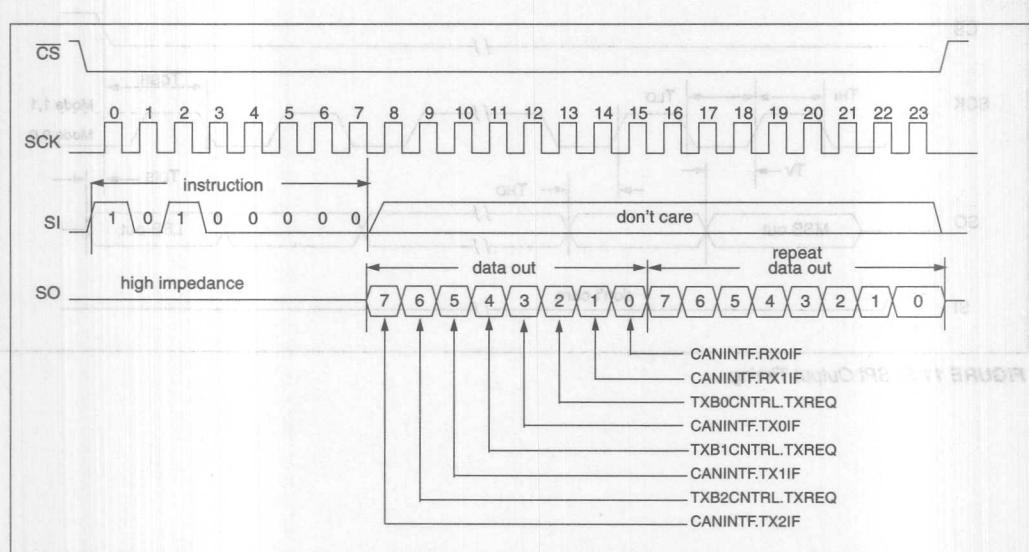


FIGURE 11-6: Read Status Instruction

MCP2510

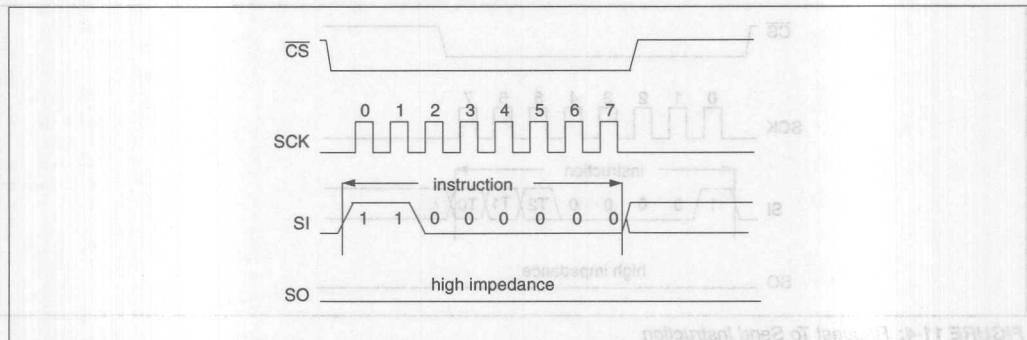


FIGURE 11-7: RESET Instruction

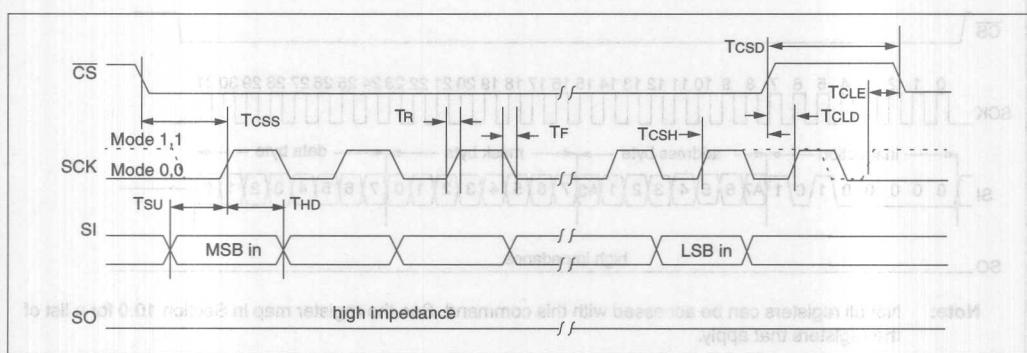


FIGURE 11-8: SPI Input Timing

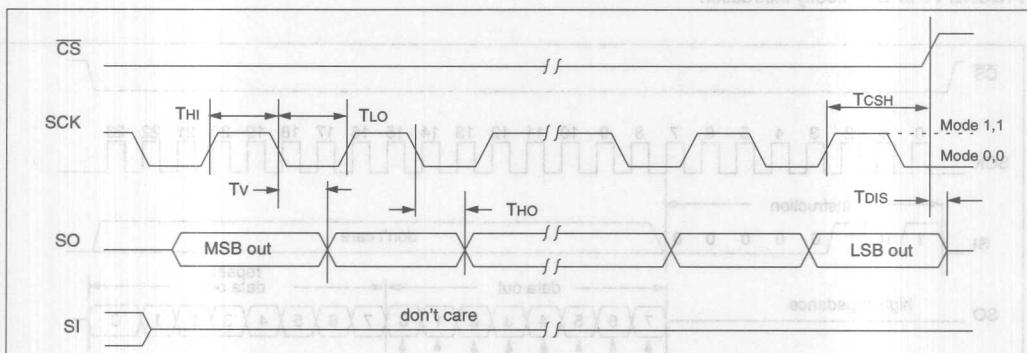


FIGURE 11-9: SPI Output Timing

12.0 ELECTRICAL CHARACTERISTICS

12.1 Maximum Ratings

V _{DD}	7.0V
All inputs and outputs w.r.t. V _{SS}	-0.6V to V _{DD} +1.0V
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	≥ 4 kV

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Parameter	Symbol	Min	Max	Units	Test Conditions
All parameters apply over the specified operating ranges unless otherwise noted.					
Supply Voltage	V _{DD}	3.0	5.5	V	
Register Retention Voltage	V _{RET}	2.4	—	V	
High Level Input Voltage					Note
RXCAN	V _{IH}	2	V _{DD} +1	V	
SCK, CS, SI, TXnRTS Pins		.7 V _{DD}	V _{DD} +1	V	
OSC1		.85 V _{DD}	V _{DD}	V	
RESET		.85 V _{DD}	V _{DD}	V	
Low Level Input Voltage					Note
RXCAN, TXnRTS Pins	V _{IL}	-0.3	.15 V _{DD}	V	
SCK, CS, SI		-0.3	.4	V	
OSC1		V _{SS}	.3 V _{DD}	V	
RESET		V _{SS}	.15 V _{DD}	V	
Low Level Output Voltage					
TXCAN	V _{OL}	—	0.4	V	I _{OL} = -6.0 mA
RXnBF Pins		—	0.4	V	I _{OL} = -8.5 mA, V _{DD} = 4.5V
SO, CLKOUT		—	0.4	V	I _{OL} = -2.1 mA, V _{DD} = 4.5V
INT		—	0.6	V	I _{OL} = -1.6 mA, V _{DD} = 4.5V
High Level Output Voltage					
TXCAN, RXnBF Pins	V _{OH}	V _{DD} -0.7	—	V	I _{OH} = 3.0mA, V _{DD} = 4.5V, I temp
SO, CLKOUT		V _{DD} -0.5	—	V	I _{OH} = 400µA
INT		V _{DD} -0.7	—	V	I _{OH} = 1.0mA, V _{DD} = 4.5V
Hysteresis	V _{HYS}	.05 V _{DD}		V	
Input Leakage Current					
All I/O except OSC1 and TXnRTS pins	I _{LI}	-1	+1	µA	CS = RESET = V _{DD} , V _{IN} = V _{SS} to V _{DD}
OSC1 Pin		-5	+5	µA	
Internal Capacitance (All Inputs And Outputs)	C _{INT}	—	7	pF	T _{AMB} = 25°C, f _C = 1.0 MHz, V _{DD} = 5.0V (Note)
Operating Current	I _{DD}	—	10	mA	V _{DD} = 5.5V; F _{CLK} =5.0 MHz; SO = Open
Standby Current (Sleep Mode)	I _{DSS}	--	5	µA	CS, TXnRTS = V _{DD} , Inputs tied to V _{DD} or V _{SS}

Note: This parameter is not 100% tested.

TABLE 12-1: DC Characteristics

All parameters apply over the specified operating ranges unless otherwise noted.	Industrial (I): $T_{AMB} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ Automotive (E): $T_{AMB} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V			
Parameter	Symbol	Min	Max	Units	Conditions
Clock In Frequency	F_{OSC}	1 1	25 16	MHz MHz	4.5V to 5.5V 3.0V to 4.5V
Clock In Period	T_{OSC}	40 62.5	1000 1000	ns ns	4.5V to 5.5V 3.0V to 4.5V
Clock In High, Low Time	T_{OSL}, T_{OSH}	10	—	ns	Note
Clock In Rise, Fall Time	T_{OSR}, T_{OSF}	—	15	ns	Note
Duty Cycle (External Clock Input)	T_{DUTY}	.30	.70	—	$T_{OSH} / (T_{OSH} + T_{OSL})$

Note: This parameter is periodically sampled and not 100% tested.

TABLE 12-2: Oscillator Timing Characteristics

All parameters apply over the specified operating ranges unless otherwise noted.	Industrial (I): $T_{AMB} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ Automotive (E): $T_{AMB} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V			
Parameter	Symbol	Min	Max	Units	Conditions
Wakeup Noise Filter	T_{WF}	50	—	ns	
CLOCKOUT Propagation Delay	T_{DCLK}	—	100	ns	

TABLE 12-3: CAN Interface AC Characteristics

All parameters apply over the specified operating ranges unless otherwise noted.	Industrial (I): $T_{AMB} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ Automotive (E): $T_{AMB} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$	$V_{CC} = 3.0\text{V}$ to 5.5V $V_{CC} = 4.5\text{V}$ to 5.5V			
Parameter	Symbol	Min	Max	Units	Conditions
CLKOUT Pin High Time	$t_{hCLKOUT}$	15	—	ns	$T_{OSC} = 40$ ns, CLKOUT prescaler set to divide by one
CLKOUT Pin Low Time	$t_{lCLKOUT}$	15	—	ns	$T_{OSC} = 40$ ns, CLKOUT prescaler set to divide by one
CLKOUT Pin Rise Time	$t_{rCLKOUT}$	—	5	ns	Measured from $0.3 V_{DD}$ to $0.7 V_{DD}$
CLKOUT Pin Fall Time	$t_{fCLKOUT}$	—	5	ns	Measured from $0.7 V_{DD}$ to $0.3 V_{DD}$
CLOCKOUT Propagation Delay	$t_{dCLKOUT}$	—	100	ns	

TABLE 12-4: CLKOUT Pin AC/DC Characteristics

Parameter	Symbol	Min	Max	Units	Conditions
CLKOUT Pin High Time	$t_{hCLKOUT}$	15	—	ns	$T_{OSC} = 40$ ns, CLKOUT prescaler set to divide by one
CLKOUT Pin Low Time	$t_{lCLKOUT}$	15	—	ns	$T_{OSC} = 40$ ns, CLKOUT prescaler set to divide by one
CLKOUT Pin Rise Time	$t_{rCLKOUT}$	—	5	ns	Measured from $0.3 V_{DD}$ to $0.7 V_{DD}$
CLKOUT Pin Fall Time	$t_{fCLKOUT}$	—	5	ns	Measured from $0.7 V_{DD}$ to $0.3 V_{DD}$
CLOCKOUT Propagation Delay	$t_{dCLKOUT}$	—	100	ns	

All parameters apply over the specified operating ranges unless otherwise noted.	Industrial (I): $T_{AMB} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ Automotive (E): $T_{AMB} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$	$V_{DD} = 3.0\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V			
Parameter	Symbol	Min	Max	Units	Test Conditions
Clock Frequency	f_{CLK}	— — —	5 4 2.5	MHz MHz MHz	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
CS Setup Time	T_{CSS}	100	—	ns	
CS Hold Time	T_{CSH}	100 115 180	— — —	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
CS Disable Time	T_{CSD}	100 100 280	— — —	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
Data Setup Time	T_{SU}	20 20 30	— — —	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
Data Hold Time	T_{HD}	20 20 50	— — —	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
CLK Rise Time	T_R	—	2	ms	Note
CLK Fall Time	T_F	—	2	ms	Note
Clock High Time	T_{HI}	90 115 180	— — —	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
Clock Low Time	T_{LO}	90 115 180	— — —	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
Clock Delay Time	T_{CLD}	50	—	ns	
Clock Enable Time	T_{CLE}	50	—	ns	
Output Valid from Clock Low	T_V	— — —	90 115 180	ns ns ns	$V_{DD} = 4.5\text{V}$ to 5.5V $V_{DD} = 4.5\text{V}$ to 5.5V (E temp) $V_{DD} = 3.0\text{V}$ to 4.5V
Output Hold Time	T_{HO}	0	—	ns	Note
Output Disable Time	T_{DIS}	—	200	ns	Note

Note: This parameter is periodically sampled and not 100% tested.

TABLE 12-5: SPI Interface AC Characteristics

MCP2510

NOTES:

13.0 PACKAGING INFORMATION

14.1 Package Marking Information

Not available at the time of printing. Will be made available after definition of QS9000 compliant standard.

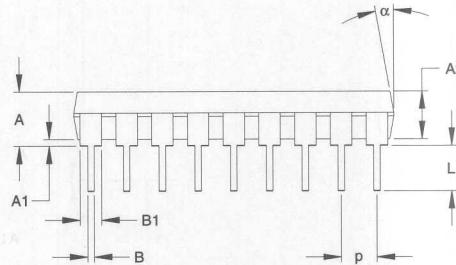
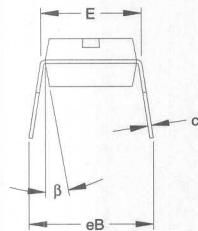
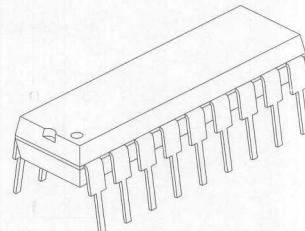
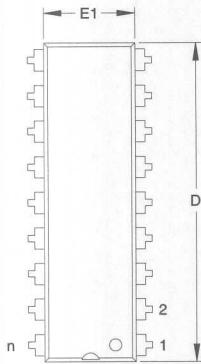
14.2 Package Details

The following sections give the technical details of the packages.

3

Datasheets

Package Type: 18-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Dimension	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	alpha	5	10	15	5	10	15
Mold Draft Angle Bottom	beta	5	10	15	5	10	15

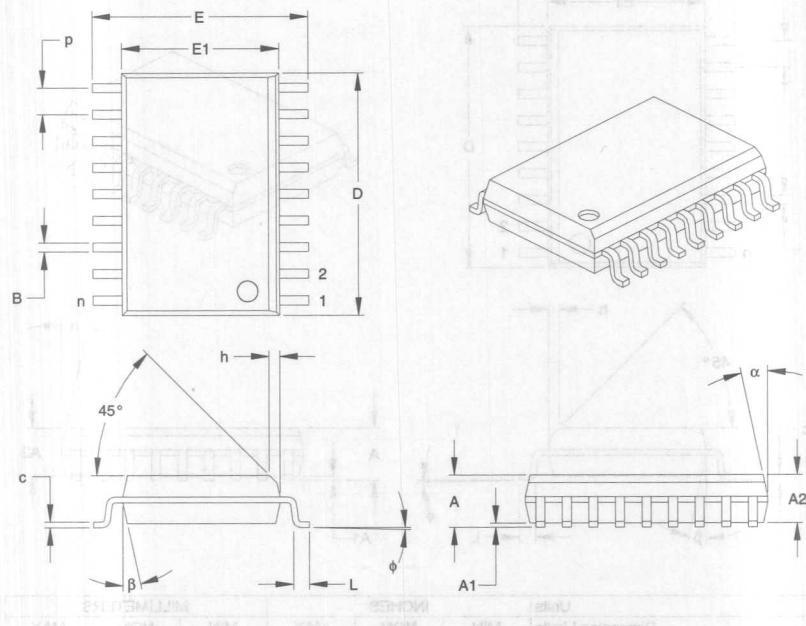
*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-007



Dimension Limits	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n			.18			.18
Pitch	p			.050			1.27
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	phi	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	alpha	0	12	15	0	12	15
Mold Draft Angle Bottom	beta	0	12	15	0	12	15

*Controlling Parameter

Notes:

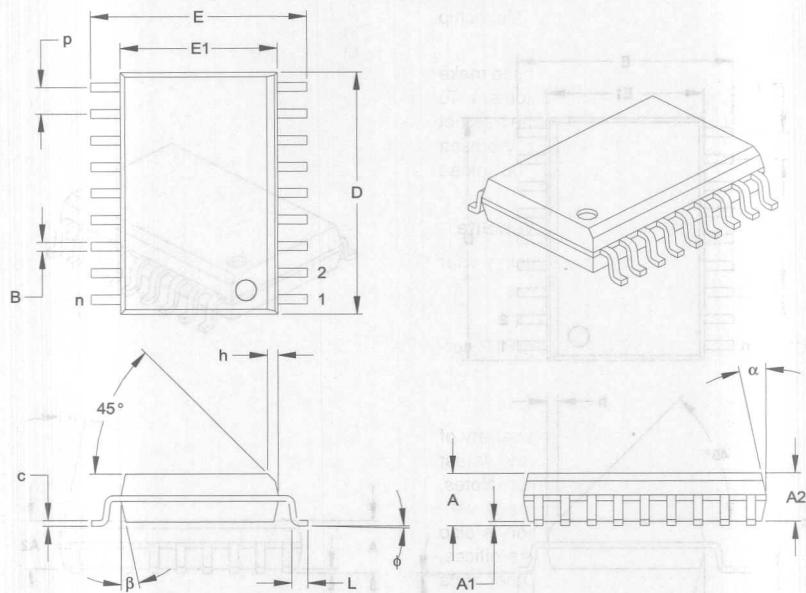
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-051

MCP2510

Package Type: 20-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP)



Dimension	Units	INCHES			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	P		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-051

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

<ftp://ftp.microchip.com>

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and
1-480-786-7302 for the rest of the world.

990902

Datasheets
Applications
Development
Tools
PICs
Data
Sheets
FAQs
Errata
Press
Releases
Technical
Support
Consultant
Program
Member
Listing
Job
Postings
Design
Tips
Device
Errata
Microchip
Consultant
Program
Member
Listing
Links
to
other
useful
web
sites
related
to
Microchip
Products
Conferences
for
products,
Development
Systems,
technical
information
and
more
Listing
of
seminars
and
events

Trademarks: The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER, PRO MATE and MPLAB are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. FlexROM and fuzzyLAB are trademarks and SQTP is a service mark of Microchip in the U.S.A.

All other trademarks mentioned herein are the property of their respective companies.

3

Datasheets

READER RESPONSE

ONLINE SURVEY

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager
RE: Reader Response 2037-88T-08A-1
From: Name
Company
Address
City / State / ZIP / Country
Telephone: () -

Total Pages Sent

Application (optional):

Would you like a reply? Y N

Device: **MCP2510**

Literature Number: **DS21291C**

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?
3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

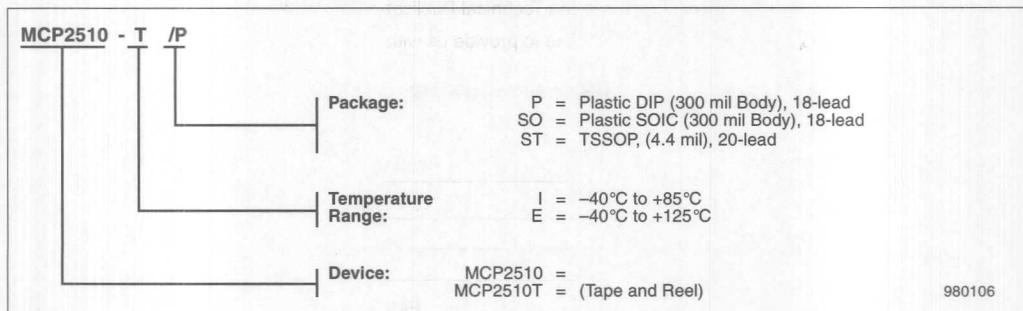
6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?

MCP2510 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277.
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCP2510

NOTES:

INDEX

Acknowledge Error	41
B	
BFpctrl - RXnBF Pin Control and Status Register	26
Bit Error	41
BIT Modify instruction	59
Bit Modify Instruction	57
Bit Timing	35
Bit Timing Configuration Registers	39
Bit Timing Logic	6
Bus Activity Wakeup Interrupt	45
Bus Off	46
Byte Write instruction	58
C	
CAN Buffers and Protocol Engine Block Diagram	5
CAN controller Register Map	55
CAN Interface AC characteristics	62
CAN Protocol Engine	6
CAN Protocol Engine Block Diagram	6
CANCTRL - CAN Control Register	52
CANINTE - Interrupt Enable Register	46
CANSTAT - CAN Status Register	53
CNF1 - Configuration Register1	39
CNF2 - Configuration Register2	40
CNF3 - Configuration Register3	40
Configuration Mode	51
CRC Error	41
Crystal/ceramic resonator operation	49
Cyclic Redundancy Check	6
D	
DC Characteristics	61
Description	1
Device Functionality	5
E	
EFLG - Error Flag Register	43
Electrical Characteristics	61
Errata	2
Error Detection	41
Error Frame	8, 13
Error Interrupt	46
Error Management Logic	6
Error Modes	42
Error Modes and Error Counters	41
Error States	41
Extended Data Frame	7
External Clock (osc1) Timing characteristics	62
External Clock Source	49
External Series Resonant Crystal Oscillator Circuit	50
F	
Features	1
Filter/Mask Truth Table	29
Form Error	41
Frame Types	7
H	
Hard Synchronization	37
I	
Information Processing Time	36
Initiating Message Transmission	15
Interframe Space	8
L	
Lengthening a Bit Period	37
Listen Only Mode	51
Loopback Mode	52
M	
Maximum Ratings	61
Message Acceptance Filter	30
Message Acceptance Filters and Masks	29
Message Reception	21
Message Reception Flowchart	23
Message Transmission	14
Modes of Operation	51
N	
Normal Mode	52
O	
Oscillator	49
Oscillator Tolerance	38
Overload Frame	8
Overview	3
P	
Package Types	1
Packaging Information	65
Phase Buffer Segments	36
Programming Time Segments	38
Propagation Segment	36
Protocol Finite State Machine	6
R	
Read Instruction	57
Read instruction Diagram	58
Read Status Instruction	57
Read Status instruction	59
REC - Receiver Error Count	42
Receive Buffers	21
Receive Buffers Diagram	22
Receive Interrupt	45
Receive Message Buffering	21
Receiver Error Passive	46
Receiver Overrun	46
Receiver Warning	46
Register Map	55
Remote Frame	7
Request To Send (RTS) Instruction	57, 59
Resynchronization	37
RXB0BF and RXB1BF Pins	21
RXB0CTRL - Receive Buffer 0 Control Register	24
RXB1CTRL - Receive Buffer 1 Control Register	25
RXBnDLC - Receive Buffer n Data Length Code	28
RXBnDm - Receive Buffer n Data Field Byte m	28
RXBnEID0 - Receive Buffer n Extended Identifier Low	27
RXBnEID8 - Receive Buffer n Extended Identifier Mid	27
RXBnSIDH - Receive Buffer n Standard Identifier High	26
RXBnSIDL - Receive Buffer n Standard Identifier Low	27
RXFnEID0 - Acceptance Filter n Extended Identifier Low	31
RXFnEID8 - Acceptance Filter n Extended Identifier Mid	31
RXFnSIDH - Acceptance Filter n Standard Identifier High	30
RXFnSIDL - Acceptance Filter n Standard Identifier Low	31
RXMnEID0 - Acceptance Filter Mask n Extended Identifier Low	33
RXMnEID8 - Acceptance Filter Mask n Extended Identifier Mid	32

MCP2510

RXMnSIDH - Acceptance Filter Mask n Standard Identifier	4
0b High	32
RXMnSIDL - Acceptance Filter Mask n Standard Identifier	4
Low	32
S	
Sample Point	36
Shortening a Bit Period	38
Sleep Mode	51
SPI Interface	57
SPI Interface Overview	57
SPI Port AC Characteristics	63
Standard Data Frame	7
Stuff Error	41
Synchronization	37
Synchronization Rules	37
Synchronization Segment	36
T	
TEC - Transmitter Error Count	42
Time Quanta	36
Transmit Interrupt	45
Transmit Message Aborting	15
Transmit Message Buffering	15
Transmit Message Buffers	15
Transmit Message flowchart	16
Transmit Message Priority	15
Transmitter Error Passive	46
Transmitter Warning	46
TXBnCTRL Transmit buffer n Control Register	17
TXBnDm - Transmit Buffer n Data Field Byte m	20
TXBnEID0 - Transmit Buffer n Extended Identifier Low	20
TXBnEID8 - Transmit Buffer n Extended Identifier Mid	19
TXBnEIDH - Transmit Buffer n Extended Identifier High	19
TXBnSIDH - Transmit Buffer n Standard Identifier High	18
TXBnSIDL - Transmit Buffer n Standard Identifier Low	19
TXnRTS Pins	15
TXRTSCTRL - TXBNRTS Pin Control and Status Register	18
U	
Typical System Implementation	4
W	
WAKE-up functions	51
Write Instruction	57
WWW, On-Line Support	2
X	
XA - Data Transfer Register	10
XB - Data Transfer Register	10
XC - Data Transfer Register	10
XD - Data Transfer Register	10
XE - Data Transfer Register	10
XF - Data Transfer Register	10
XA - Data Transfer Register	10
XB - Data Transfer Register	10
XC - Data Transfer Register	10
XD - Data Transfer Register	10
XE - Data Transfer Register	10
XF - Data Transfer Register	10
Z	
ZA - Data Transfer Register	10
ZB - Data Transfer Register	10
ZC - Data Transfer Register	10
ZD - Data Transfer Register	10
ZE - Data Transfer Register	10
ZF - Data Transfer Register	10
E	
EA - Data Transfer Register	10
EB - Data Transfer Register	10
EC - Data Transfer Register	10
ED - Data Transfer Register	10
EE - Data Transfer Register	10
EF - Data Transfer Register	10
F	
FA - Data Transfer Register	10
FB - Data Transfer Register	10
FC - Data Transfer Register	10
FD - Data Transfer Register	10
FE - Data Transfer Register	10
FF - Data Transfer Register	10
G	
GA - Data Transfer Register	10
GB - Data Transfer Register	10
GC - Data Transfer Register	10
GD - Data Transfer Register	10
GE - Data Transfer Register	10
GF - Data Transfer Register	10
H	
HA - Data Transfer Register	10
HB - Data Transfer Register	10
HC - Data Transfer Register	10
HD - Data Transfer Register	10
HE - Data Transfer Register	10
HF - Data Transfer Register	10
I	
IA - Data Transfer Register	10
IB - Data Transfer Register	10
IC - Data Transfer Register	10
ID - Data Transfer Register	10
IE - Data Transfer Register	10
IF - Data Transfer Register	10
J	
JA - Data Transfer Register	10
JB - Data Transfer Register	10
JC - Data Transfer Register	10
JD - Data Transfer Register	10
JE - Data Transfer Register	10
JF - Data Transfer Register	10
K	
KA - Data Transfer Register	10
KB - Data Transfer Register	10
KC - Data Transfer Register	10
KD - Data Transfer Register	10
KE - Data Transfer Register	10
KF - Data Transfer Register	10
L	
LA - Data Transfer Register	10
LB - Data Transfer Register	10
LC - Data Transfer Register	10
LD - Data Transfer Register	10
LE - Data Transfer Register	10
LF - Data Transfer Register	10
M	
MA - Data Transfer Register	10
MB - Data Transfer Register	10
MC - Data Transfer Register	10
MD - Data Transfer Register	10
ME - Data Transfer Register	10
MF - Data Transfer Register	10
N	
NA - Data Transfer Register	10
NB - Data Transfer Register	10
NC - Data Transfer Register	10
ND - Data Transfer Register	10
NE - Data Transfer Register	10
NF - Data Transfer Register	10
P	
PA - Data Transfer Register	10
PB - Data Transfer Register	10
PC - Data Transfer Register	10
PD - Data Transfer Register	10
PE - Data Transfer Register	10
PF - Data Transfer Register	10
R	
RA - Data Transfer Register	10
RB - Data Transfer Register	10
RC - Data Transfer Register	10
RD - Data Transfer Register	10
RE - Data Transfer Register	10
RF - Data Transfer Register	10
S	
SA - Data Transfer Register	10
SB - Data Transfer Register	10
SC - Data Transfer Register	10
SD - Data Transfer Register	10
SE - Data Transfer Register	10
SF - Data Transfer Register	10
T	
TA - Data Transfer Register	10
TB - Data Transfer Register	10
TC - Data Transfer Register	10
TD - Data Transfer Register	10
TE - Data Transfer Register	10
TF - Data Transfer Register	10
U	
UA - Data Transfer Register	10
UB - Data Transfer Register	10
UC - Data Transfer Register	10
UD - Data Transfer Register	10
UE - Data Transfer Register	10
UF - Data Transfer Register	10
V	
VA - Data Transfer Register	10
VB - Data Transfer Register	10
VC - Data Transfer Register	10
VD - Data Transfer Register	10
VE - Data Transfer Register	10
VF - Data Transfer Register	10
W	
WA - Data Transfer Register	10
WB - Data Transfer Register	10
WC - Data Transfer Register	10
WD - Data Transfer Register	10
WE - Data Transfer Register	10
WF - Data Transfer Register	10
X	
XA - Data Transfer Register	10
XB - Data Transfer Register	10
XC - Data Transfer Register	10
XD - Data Transfer Register	10
XE - Data Transfer Register	10
XF - Data Transfer Register	10
Y	
YA - Data Transfer Register	10
YB - Data Transfer Register	10
YC - Data Transfer Register	10
YD - Data Transfer Register	10
YE - Data Transfer Register	10
YF - Data Transfer Register	10
Z	
ZA - Data Transfer Register	10
ZB - Data Transfer Register	10
ZC - Data Transfer Register	10
ZD - Data Transfer Register	10
ZE - Data Transfer Register	10
ZF - Data Transfer Register	10

LIST OF FIGURES

Figure 1-1:	Block Diagram	3
Figure 1-2:	Typical System Implementation	4
Figure 1-3:	CAN Buffers and Protocol Engine Block Diagram	5
Figure 1-4:	CAN Protocol Engine Block Diagram	6
Figure 2-1:	Standard Data Frame	9
Figure 2-2:	Extended Data Frame	10
Figure 2-3:	Remote Data Frame	11
Figure 2-4:	Error Frame	12
Figure 2-5:	Overload Frame	13
Figure 3-1:	Transmit Message Flowchart	16
Figure 4-1:	Receive Buffer Block Diagram	22
Figure 4-2:	Message Reception Flowchart	23
Figure 4-3:	Message Acceptance Mask and Filter Operation	30
Figure 5-1:	Bit Time Partitioning	35
Figure 5-2:	Lengthening a Bit Period	37
Figure 5-3:	Shortening a Bit Period	38
Figure 6-1:	Error Modes State Diagram	42
Figure 8-1:	Crystal/Ceramic Resonator Operation	49
Figure 8-2:	External Clock Source	49
Figure 8-3:	External Series Resonant Crystal Oscillator Circuit	50
Figure 11-1:	Bit Modify	58
Figure 11-2:	Read Instruction	58
Figure 11-3:	Byte Write Instruction	58
Figure 11-4:	Request To Send Instruction	59
Figure 11-5:	BIT Modify instruction	59
Figure 11-6:	Read Status Instruction	59
Figure 11-7:	RESET Instruction	60
Figure 11-8:	SPI Input Timing	60
Figure 11-9:	SPI Output Timing	60

LIST OF TABLES

Table 1-1:	Pin Descriptions	4
Table 4-10:	Filter/Mask Truth Table	29
Table 7-1:	ICOD<2:0> Decode	45
Table 10-1:	CAN Controller Register Map	55
Table 10-2:	Control Register Summary	55
Table 11-1:	SPI Instruction Set	58
Table 12-1:	DC Characteristics	61
Table 12-2:	Oscillator Timing Characteristics	62
Table 12-3:	CAN Interface AC Characteristics	62
Table 12-4:	CLKOUT Pin AC/DC Characteristics	62
Table 12-5:	SPI Interface AC Characteristics	63

LIST OF REGISTERS

Register 3-1:	TXBnCTRL Transmit Buffer n Control Register	17
Register 3-2:	TXRTSCTRL - TXnRTS Pin Control and Status Register	18
Register 3-3:	TXBnSIDH - Transmit Buffer n Standard Identifier High	18
Register 3-4:	TXBnSIDL - Transmit Buffer n Standard Identifier Low	19
Register 3-5:	TXBnEID8 - Transmit Buffer n Extended Identifier High	19
Register 3-6:	TXBnEID0 - Transmit Buffer n Extended Identifier LOW	19
Register 3-7:	TXBnDLC - Transmit Buffer n Data Length Code	20
Register 3-8:	TXBnDm - Transmit Buffer n Data Field Byte m	20
Register 4-1:	RXB0CTRL - Receive Buffer 0 Control Register	24
Register 4-2:	RXB1CTRL - Receive Buffer 1 Control Register	25
Register 4-3:	BFPCTRL - RXnBF Pin Control and Status Register	26
Register 4-4:	RXBnSIDH - Receive Buffer n Standard Identifier High	26
Register 4-5:	RXBnSIDL - Receive Buffer n Standard Identifier Low	27
Register 4-6:	RXBnEID8 - Receive Buffer n Extended Identifier Mid	27
Register 4-7:	RXBnEID0 - Receive Buffer n Extended Identifier Low	27
Register 4-8:	RXBnDLC - Receive Buffer n Data Length Code	28
Register 4-9:	RXBnDM - Receive Buffer n Data Field Byte m	28
Register 4-10:	RXFnSIDH - Acceptance Filter n Standard Identifier High	30
Register 4-11:	RXFnSIDL - Acceptance Filter n Standard Identifier Low	31
Register 4-12:	RXMnEID8 - Acceptance Filter n Extended Identifier High	31
Register 4-13:	RXMnEID0 - Acceptance Filter n Extended Identifier Low	31
Register 4-14:	RXMnSIDH - Acceptance Filter Mask n Standard Identifier High	32
Register 4-15:	RXMnSIDL - Acceptance Filter Mask n Standard Identifier Low	32
Register 4-16:	RXMnEID8 - Acceptance Filter Mask n Extended Identifier High	32
Register 4-17:	RXMnEID0 - Acceptance Filter Mask n Extended Identifier Low	33
Register 5-1:	CNF1 - Configuration Register1	39
Register 5-2:	CNF2 - Configuration Register2	40
Register 5-3:	CNF3 - Configuration Register3	40
Register 6-1:	TEC - Transmitter Error Counter	42
Register 6-2:	REC - Receiver Error Counter	42
Register 6-3:	EFLG - Error Flag Register	43
Register 7-1:	CANINTE - Interrupt Enable Register ..	46
Register 7-2:	CANINTF - Interrupt FLAG Register ..	47
Register 9-1:	CANCTRL - CAN Control Register ..	52
Register 9-2:	CANSTAT - CAN Status Register ..	53

NOTES:



	LIST OF REGISTERS	1
15	TX/RX Control, Transmit Buffer Descriptor	17
16	Registers 3-17	17
17	TX/RX Control, Receive Buffer Descriptor	18
18	Registers 3-18	18
19	TX/RX Control, Transfer Buffer	19
20	Registers 3-19	19
21	TX/RX Control, Transfer Buffer	20
22	Registers 3-20	20
23	TX/RX Control, Transfer Buffer	21
24	Registers 3-21	21
25	TX/RX Control, Transfer Buffer	22
26	Registers 3-22	22
27	TX/RX Control, Transfer Buffer	23
28	Registers 3-23	23
29	TX/RX Control, Transfer Buffer	24
30	Registers 3-24	24
31	RX Buffer Descriptor	25
32	Registers 4-1	25
33	TX Buffer Descriptor	26
34	Registers 4-2	26
35	TX Buffer Descriptor	27
36	Registers 4-3	27
37	TX Buffer Descriptor	28
38	Registers 4-4	28
39	TX Buffer Descriptor	29
40	Registers 4-5	29
41	TX Buffer Descriptor	30
42	Registers 4-6	30
43	TX Buffer Descriptor	31
44	Registers 4-7	31
45	TX Buffer Descriptor	32
46	Registers 4-8	32
47	TX Buffer Descriptor	33
48	Registers 4-9	33
49	TX Buffer Descriptor	34
50	Registers 4-10	34
51	TX Buffer Descriptor	35
52	Registers 4-11	35
53	TX Buffer Descriptor	36
54	Registers 4-12	36
55	TX Buffer Descriptor	37
56	Registers 4-13	37
57	TX Buffer Descriptor	38
58	Registers 4-14	38
59	TX Buffer Descriptor	39
60	Registers 4-15	39
61	TX Buffer Descriptor	40
62	Registers 4-16	40
63	TX Buffer Descriptor	41
64	Registers 4-17	41
65	TX Buffer Descriptor	42
66	Registers 4-18	42
67	TX Buffer Descriptor	43
68	Registers 4-19	43
69	TX Buffer Descriptor	44
70	Registers 4-20	44
71	TX Buffer Descriptor	45
72	Registers 4-21	45
73	TX Buffer Descriptor	46
74	Registers 4-22	46
75	TX Buffer Descriptor	47
76	Registers 4-23	47
77	TX Buffer Descriptor	48
78	Registers 4-24	48
79	TX Buffer Descriptor	49
80	Registers 4-25	49
81	TX Buffer Descriptor	50
82	Registers 4-26	50
83	TX Buffer Descriptor	51
84	Registers 4-27	51
85	TX Buffer Descriptor	52
86	Registers 4-28	52
87	TX Buffer Descriptor	53
88	Registers 4-29	53
89	TX Buffer Descriptor	54
90	Registers 4-30	54
91	TX Buffer Descriptor	55
92	Registers 4-31	55
93	TX Buffer Descriptor	56
94	Registers 4-32	56
95	TX Buffer Descriptor	57
96	Registers 4-33	57
97	TX Buffer Descriptor	58
98	Registers 4-34	58
99	TX Buffer Descriptor	59
100	Registers 4-35	59
101	TX Buffer Descriptor	60
102	Registers 4-36	60
103	TX Buffer Descriptor	61
104	Registers 4-37	61
105	TX Buffer Descriptor	62
106	Registers 4-38	62
107	TX Buffer Descriptor	63
108	Registers 4-39	63
109	TX Buffer Descriptor	64
110	Registers 4-40	64
111	TX Buffer Descriptor	65
112	Registers 4-41	65
113	TX Buffer Descriptor	66
114	Registers 4-42	66
115	TX Buffer Descriptor	67
116	Registers 4-43	67
117	TX Buffer Descriptor	68
118	Registers 4-44	68
119	TX Buffer Descriptor	69
120	Registers 4-45	69
121	TX Buffer Descriptor	70
122	Registers 4-46	70
123	TX Buffer Descriptor	71
124	Registers 4-47	71
125	TX Buffer Descriptor	72
126	Registers 4-48	72
127	TX Buffer Descriptor	73
128	Registers 4-49	73
129	TX Buffer Descriptor	74
130	Registers 4-50	74
131	TX Buffer Descriptor	75
132	Registers 4-51	75
133	TX Buffer Descriptor	76
134	Registers 4-52	76
135	TX Buffer Descriptor	77
136	Registers 4-53	77
137	TX Buffer Descriptor	78
138	Registers 4-54	78
139	TX Buffer Descriptor	79
140	Registers 4-55	79
141	TX Buffer Descriptor	80
142	Registers 4-56	80
143	TX Buffer Descriptor	81
144	Registers 4-57	81
145	TX Buffer Descriptor	82
146	Registers 4-58	82
147	TX Buffer Descriptor	83
148	Registers 4-59	83
149	TX Buffer Descriptor	84
150	Registers 4-60	84
151	TX Buffer Descriptor	85
152	Registers 4-61	85
153	TX Buffer Descriptor	86
154	Registers 4-62	86
155	TX Buffer Descriptor	87
156	Registers 4-63	87
157	TX Buffer Descriptor	88
158	Registers 4-64	88
159	TX Buffer Descriptor	89
160	Registers 4-65	89
161	TX Buffer Descriptor	90
162	Registers 4-66	90
163	TX Buffer Descriptor	91
164	Registers 4-67	91
165	TX Buffer Descriptor	92
166	Registers 4-68	92
167	TX Buffer Descriptor	93
168	Registers 4-69	93
169	TX Buffer Descriptor	94
170	Registers 4-70	94
171	TX Buffer Descriptor	95
172	Registers 4-71	95
173	TX Buffer Descriptor	96
174	Registers 4-72	96
175	TX Buffer Descriptor	97
176	Registers 4-73	97
177	TX Buffer Descriptor	98
178	Registers 4-74	98
179	TX Buffer Descriptor	99
180	Registers 4-75	99
181	TX Buffer Descriptor	100
182	Registers 4-76	100
183	TX Buffer Descriptor	101
184	Registers 4-77	101
185	TX Buffer Descriptor	102
186	Registers 4-78	102
187	TX Buffer Descriptor	103
188	Registers 4-79	103
189	TX Buffer Descriptor	104
190	Registers 4-80	104
191	TX Buffer Descriptor	105
192	Registers 4-81	105
193	TX Buffer Descriptor	106
194	Registers 4-82	106
195	TX Buffer Descriptor	107
196	Registers 4-83	107
197	TX Buffer Descriptor	108
198	Registers 4-84	108
199	TX Buffer Descriptor	109
200	Registers 4-85	109
201	TX Buffer Descriptor	110
202	Registers 4-86	110
203	TX Buffer Descriptor	111
204	Registers 4-87	111
205	TX Buffer Descriptor	112
206	Registers 4-88	112
207	TX Buffer Descriptor	113
208	Registers 4-89	113
209	TX Buffer Descriptor	114
210	Registers 4-90	114
211	TX Buffer Descriptor	115
212	Registers 4-91	115
213	TX Buffer Descriptor	116
214	Registers 4-92	116
215	TX Buffer Descriptor	117
216	Registers 4-93	117
217	TX Buffer Descriptor	118
218	Registers 4-94	118
219	TX Buffer Descriptor	119
220	Registers 4-95	119
221	TX Buffer Descriptor	120
222	Registers 4-96	120
223	TX Buffer Descriptor	121
224	Registers 4-97	121
225	TX Buffer Descriptor	122
226	Registers 4-98	122
227	TX Buffer Descriptor	123
228	Registers 4-99	123
229	TX Buffer Descriptor	124
230	Registers 4-100	124
231	TX Buffer Descriptor	125
232	Registers 4-101	125
233	TX Buffer Descriptor	126
234	Registers 4-102	126
235	TX Buffer Descriptor	127
236	Registers 4-103	127
237	TX Buffer Descriptor	128
238	Registers 4-104	128
239	TX Buffer Descriptor	129
240	Registers 4-105	129
241	TX Buffer Descriptor	130
242	Registers 4-106	130
243	TX Buffer Descriptor	131
244	Registers 4-107	131
245	TX Buffer Descriptor	132
246	Registers 4-108	132
247	TX Buffer Descriptor	133
248	Registers 4-109	133
249	TX Buffer Descriptor	134
250	Registers 4-110	134
251	TX Buffer Descriptor	135
252	Registers 4-111	135
253	TX Buffer Descriptor	136
254	Registers 4-112	136
255	TX Buffer Descriptor	137
256	Registers 4-113	137
257	TX Buffer Descriptor	138
258	Registers 4-114	138
259	TX Buffer Descriptor	139
260	Registers 4-115	139
261	TX Buffer Descriptor	140
262	Registers 4-116	140
263	TX Buffer Descriptor	141
264	Registers 4-117	141
265	TX Buffer Descriptor	142
266	Registers 4-118	142
267	TX Buffer Descriptor	143
268	Registers 4-119	143
269	TX Buffer Descriptor	144
270	Registers 4-120	144
271	TX Buffer Descriptor	145
272	Registers 4-121	145
273	TX Buffer Descriptor	146
274	Registers 4-122	146
275	TX Buffer Descriptor	147
276	Registers 4-123	147
277	TX Buffer Descriptor	148
278	Registers 4-124	148
279	TX Buffer Descriptor	149
280	Registers 4-125	149
281	TX Buffer Descriptor	150
282	Registers 4-126	150
283	TX Buffer Descriptor	151
284	Registers 4-127	151
285	TX Buffer Descriptor	152
286	Registers 4-128	152
287	TX Buffer Descriptor	153
288	Registers 4-129	153
289	TX Buffer Descriptor	154
290	Registers 4-130	154
291	TX Buffer Descriptor	155
292	Registers 4-131	155
293	TX Buffer Descriptor	156
294	Registers 4-132	156
295	TX Buffer Descriptor	157
296	Registers 4-133	157
297	TX Buffer Descriptor	158
298	Registers 4-134	158
299	TX Buffer Descriptor	159
300	Registers 4-135	159
301	TX Buffer Descriptor	160
302	Registers 4-136	160
303	TX Buffer Descriptor	161
304	Registers 4-137	161
305	TX Buffer Descriptor	162
306	Registers 4-138	162
307	TX Buffer Descriptor	163
308	Registers 4-139	163
309	TX Buffer Descriptor	164
310	Registers 4-140	164
311	TX Buffer Descriptor	165
312	Registers 4-141	165
313	TX Buffer Descriptor	166
314	Registers 4-142	166
315	TX Buffer Descriptor	167
316	Registers 4-143	167
317	TX Buffer Descriptor	168
318	Registers 4-144	168
319	TX Buffer Descriptor	169
320	Registers 4-145	169
321	TX Buffer Descriptor	170
322	Registers 4-146	170
323	TX Buffer Descriptor	171
324	Registers 4-147	171
325	TX Buffer Descriptor	172
326	Registers 4-148	172
327	TX Buffer Descriptor	173
328	Registers 4-149	173
329	TX Buffer Descriptor	174
330	Registers 4-150	174
331	TX Buffer Descriptor	175
332	Registers 4-151	175
333	TX Buffer Descriptor	176
334	Registers 4-152	176
335	TX Buffer Descriptor	177
336	Registers 4-153	177
337	TX Buffer Descriptor	178
338	Registers 4-154	178
339	TX Buffer Descriptor	179
340	Registers 4-155	179
341	TX Buffer Descriptor	180
342	Registers 4-156	180
343	TX Buffer Descriptor	181
344	Registers 4-157	181
345	TX Buffer Descriptor	182
346	Registers 4-158	182
347	TX Buffer Descriptor	183
348	Registers 4-159	183
349	TX Buffer Descriptor	184
350	Registers 4-160	184
351	TX Buffer Descriptor	185
352	Registers 4-161	185
353	TX Buffer Descriptor	186
354	Registers 4-162	186
355	TX Buffer Descriptor	187
356	Registers 4-163	187
357	TX Buffer Descriptor	188
358	Registers 4-164	188
359	TX Buffer Descriptor	189
360	Registers 4-165	189
361	TX Buffer Descriptor	190
362	Registers 4-166	190
363	TX Buffer Descriptor	191
364	Registers 4-167	191
365	TX Buffer Descriptor	192
366	Registers 4-168	192
367	TX Buffer Descriptor	193
368	Registers 4-169	193
369	TX Buffer Descriptor	194
370	Registers 4-170	194
371	TX Buffer Descriptor	195
372	Registers 4-171	195
373	TX Buffer Descriptor	196
374	Registers 4-172	196
375	TX Buffer Descriptor	197
376	Registers 4-173	197
377	TX Buffer Descriptor	198
378	Registers 4-174	198
379	TX Buffer Descriptor	199
380	Registers 4-175	199
381	TX Buffer Descriptor	200
382	Registers 4-176	200
383	TX Buffer Descriptor	201
384	Registers 4-177	201
385	TX Buffer Descriptor	202
386	Registers 4-178	202
387	TX Buffer Descriptor	203
388	Registers 4-179	203
389	TX Buffer Descriptor	204
390	Registers 4-180	204
391	TX Buffer Descriptor	205
392	Registers 4-181	205
393	TX Buffer Descriptor	206
394	Registers 4-182	206
395	TX Buffer Descriptor	207
396	Registers 4-183	207
397	TX Buffer Descriptor	208
398	Registers 4-184	208
399	TX Buffer Descriptor	209
400	Registers 4-185	209
401	TX Buffer Descriptor	210
402	Registers 4-186	210
403	TX Buffer Descriptor	211
404	Registers 4-187	211
405	TX Buffer Descriptor	212
406	Registers 4-188	212
407	TX Buffer Descriptor	213
408	Registers 4-189	213
409	TX Buffer Descriptor	214
410	Registers 4-190	214
411	TX Buffer Descriptor	215
412	Registers 4-191	215
413	TX Buffer Descriptor	216
414	Registers 4-192	216
415	TX Buffer Descriptor	217
416	Registers 4-193	217
417	TX Buffer Descriptor	218
418	Registers 4-194	218
419	TX Buffer Descriptor	219
420	Registers 4-195	219
421	TX Buffer Descriptor	220
422	Registers 4-196	220
423	TX Buffer Descriptor	221
424	Registers 4-197	221
425	TX Buffer Descriptor	222
426	Registers 4-198	222
427	TX Buffer Descriptor	223
428	Registers 4-199	223
429	TX Buffer Descriptor	224
430	Registers 4-200	

2.7V to 5.5V Single Supply CMOS Op Amps

FEATURES

- Specifications rated from 2.7V to 5.5V supplies
- Rail-to-rail swing at output
- Common-mode input swing below ground
- 2.8MHz GBWP
- Unity gain stable
- Low power $I_{DD} = 325\mu A$ max
- Chip Select capability with MCP603
- Industrial temperature range (-40°C to 85°C)
- Available in single, dual and quad

APPLICATIONS

- Portable Equipment
- A/D Converter Driver
- Photodiode Pre-amps
- Analog Filters
- Data Acquisition
- Notebooks and PDAs
- Sensor Interface

AVAILABLE TOOLS

- Spice Macromodels (at www.microchip.com)
 - **FilterLab™** Software (at www.microchip.com)
- © 2000 Microchip Technology Inc.

DESCRIPTION

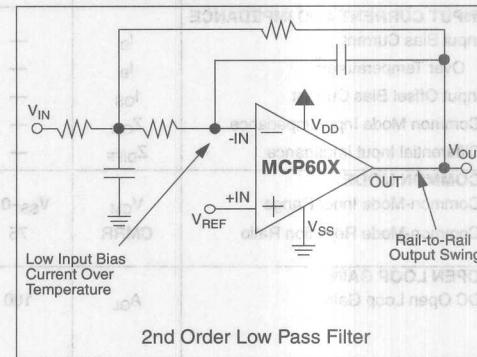
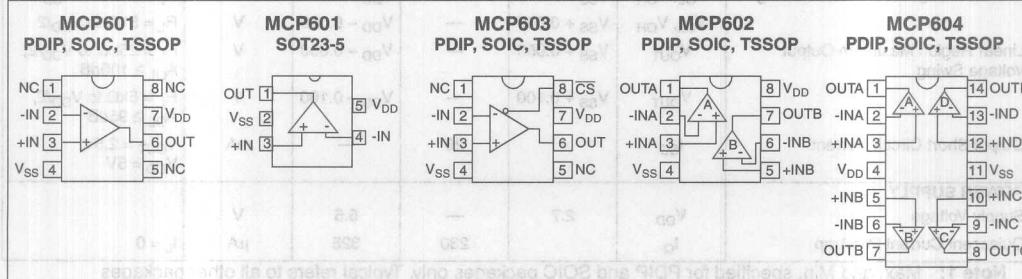
The Microchip Technology Inc. MCP601/602/603/604 family of low power operational amplifiers are offered in single (MCP601), single with a Chip Select pin feature (MCP603), dual (MCP602) and quad (MCP604) configurations. These operational amplifiers (op amps) utilize an advanced CMOS technology, which provides low bias current, high speed operation, high open-loop gain and rail-to-rail output swing. This product offering oper-

Amplifiers

ates with a single supply voltage that can be as low as 2.7V, while drawing less than 325 μA of quiescent current. In addition, the common-mode input voltage range goes 0.3V below ground, making these amplifiers ideal for single supply operation.

These devices are appropriate for low-power battery operated circuits due to the low quiescent current, for A/D Converter driver amplifiers because of their wide bandwidth, or for anti-aliasing filters by virtue of their low input bias current.

The MCP601, MCP602 and MCP603 are available in standard 8-lead PDIP, SOIC and TSSOP packages. The MCP601 is also available in the SOT23-5 package. The quad MCP604 is offered in 14-lead PDIP, SOIC and TSSOP packages. PDIP and SOIC packages are fully specified from -40°C to +85°C with power supplies from 2.7V to 5.5V.

TYPICAL APPLICATION**PACKAGES**

MCP601/602/603/604

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t.	V_{SS} -0.3V to V_{DD} +0.3V
Difference Input voltage	$ V_{DD} - V_{SS} $
Output Short Circuit Current	continuous
Current at Input Pin	±2mA
Current at Output and Supply Pins	±30mA
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-55°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD Tolerance	3kV Human Body Model

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

Unless otherwise indicated, all limits are specified for $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 100k\Omega$ to $V_{DD}/2$, and $V_{OUT} \sim V_{DD}/2$.

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
INPUT OFFSET VOLTAGE						
Input Offset Voltage	V_{OS}	-2		+2	mV	
Over Temperature ⁽¹⁾	V_{OS}	-3		+3	mV	$T_A = -40^\circ C$ to $+85^\circ C$
Drift with Temperature	dV_{OS}/dT	—	±2.5	—	$\mu V/^\circ C$	$T_A = -40^\circ C$ to $+85^\circ C$
Power Supply Rejection	PSRR	—	40	100	$\mu V/V$	for $V_{DD} = 2.7V$ to $5.5V$
INPUT CURRENT AND IMPEDANCE						
Input Bias Current	I_B	—	1	—	pA	
Over Temperature ⁽²⁾	I_B	—	20	60	pA	$T_A = -40^\circ C$ to $+85^\circ C$
Input Offset Bias Current	I_{OS}	—	1	—	pA	
Common Mode Input Impedance	Z_{CM}	—	10^{13} to 16	—	$\Omega l/pF$	
Differential Input Impedance	Z_{DIFF}	—	10^{13} to 13	—	$\Omega l/pF$	
COMMON MODE						
Common-Mode Input Range	V_{CM}	$V_{SS} - 0.3$	—	$V_{DD} - 1.2$	V	
Common-Mode Rejection Ratio	CMRR	75	90 to 800	—	dB	$V_{DD} = 5V$, $V_{CM} = -0.3$ to $3.8V$
OPEN LOOP GAIN						
DC Open Loop Gain	A_{OL}	100	115	—	dB	$R_L = 25k\Omega$ to $V_{DD}/2$, $50mV < V_{OUT} < (V_{DD} - 50 mV)$
DC Open Loop Gain	A_{OL}	95	110	—	dB	$R_L = 5k\Omega$ to $V_{DD}/2$, $100mV < V_{OUT} < (V_{DD} - 100mV)$
OUTPUT						
Low Level/High Level Output Swing	V_{OL} , V_{OH}	$V_{SS} + 0.015$	—	$V_{DD} - 0.020$	V	$R_L = 25k\Omega$ to $V_{DD}/2$
V_{OL} , V_{OH}	$V_{SS} + 0.045$	—	$V_{DD} - 0.060$	V		$R_L = 5k\Omega$ to $V_{DD}/2$
V_{OUT}	$V_{SS} + 0.050$	—	$V_{DD} - 0.050$	V		$R_L = 25k\Omega$ to $V_{DD}/2$, $A_{OL} \geq 100dB$
Linear Region Maximum Output Voltage Swing	V_{OUT}	$V_{SS} + 0.100$	—	$V_{DD} - 0.100$	V	$R_L = 5k\Omega$ to $V_{DD}/2$, $A_{OL} \geq 95dB$
Output Short Circuit Current	I_{SC}	20	—	—	mA	$V_{OUT} = 2.5V$, $V_{DD} = 5V$
POWER SUPPLY						
Supply Voltage	V_{DD}	2.7	—	5.5	V	
Quiescent Current Per Amp	I_Q	230		325	μA	$I_L = 0$

Note 1: Max. and Min. specified for PDIP and SOIC packages only. Typical refers to all other packages

Note 2: Max. and Min. specified for PDIP, SOIC, and TSSOP packages only. Typical refers to all packages.

PIN FUNCTION TABLE

NAME	FUNCTION
+IN, +INA, +INB, +INC, +IND	Non-inverting Input Terminals
-IN, -INA, -INB, -INC, -IND	Inverting Input Terminals
V_{DD}	Positive Power Supply
V_{SS}	Negative Power Supply
OUT, OUTA, OUTB, OUTC, OUTD	Output Terminals
CS	Chip Select
NC	No internal connection to IC

AC CHARACTERISTICS

Unless otherwise indicated, all limits are specified for $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 100k\Omega$ to $V_{DD}/2$, and $V_{OUT} \sim V_{DD}/2$

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Gain Bandwidth Product	GBWP	—	2.8	—	MHz	$V_{DD} = 5V$
Phase Margin	Θ_m	—	50	—	degrees	$C_L = 50pF$, $V_{DD} = 5V$
Slew Rate	SR	—	2.3	—	V/ μ s	$G = +1V/V$, $V_{DD} = 5V$
Setting Time to 0.01%		—	4.5	—	μ s	for $\Delta V_{OUT} = 3.8V$ STEP, $C_L = 50pF$, $V_{DD} = 5V$, $G = +1V/V$
NOISE						
Input Voltage Noise	e_n	—	7	—	μV_{P-P}	$f = 0.1Hz$ to $10Hz$
Input Voltage Noise Density	e_n	—	29	—	nV/ \sqrt{Hz}	$f = 1kHz$
Input Current Noise Density	i_n	—	0.6	—	fA/ \sqrt{Hz}	$f = 1kHz$

SPECIFICATIONS FOR MCP603 CHIP SELECT FEATURE

Unless otherwise indicated, all limits are specified for $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 100k\Omega$ to $V_{DD}/2$, and $V_{OUT} \sim V_{DD}/2$

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
CS LOW SPECIFICATIONS						
CS Logic Threshold, Low	V_{IL}	V_{SS}	0.42 V_{DD}	0.2 V_{DD}	V	For entire V_{DD} range
CS Input Current, Low	I_{CSL}	-1.0	—	—	μA	$\overline{CS} = 0.2V_{DD}$
Amplifier Output Leakage, CS High		—	1	—	nA	
CS HIGH SPECIFICATIONS						
CS Logic Threshold, High	V_{IH}	0.8 V_{DD}	0.51 V_{DD}	V_{DD}	V	For entire V_{DD} range
CS Input High, Shutdown CS Pin Current	I_{CSH}	—	0.7	2.0	μA	$\overline{CS} = V_{DD}$
CS Input High, Shutdown GND Current	I_Q	—	0.7	2.0	μA	$\overline{CS} = V_{DD}$
DYNAMIC SPECIFICATIONS						
CS Low to Amplifier Output High Turn-on Time	t_{ON}	—	3.1	10	μs	\overline{CS} low $\leq 0.2V_{DD}$
CS High to Amplifier Output High Z	t_{OFF}	—	100	—	ns	\overline{CS} high $\geq 0.8V_{DD}$, No Load
CS Threshold Hysteresis		—	0.3	—	V	

TEMPERATURE SPECIFICATIONS

Unless otherwise indicated, all limits are specified for $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TEMPERATURE RANGE						
Specified Temperature Range	T_A	-40	—	+85	$^\circ C$	
Operating Temperature Range	T_A	-40	—	+85	$^\circ C$	
Storage Temperature Range	T_A	-65	—	+150	$^\circ C$	
THERMAL PACKAGE RESISTANCE						
Thermal Resistance, 5L-SOT23-5	θ_{JA}	—	256	—	$^\circ C/W$	
Thermal Resistance, 8L-PDIP	θ_{JA}	—	85	—	$^\circ C/W$	
Thermal Resistance, 8L-SOIC	θ_{JA}	—	163	—	$^\circ C/W$	
Thermal Resistance, 8L-TSSOP	θ_{JA}	—	124	—	$^\circ C/W$	
Thermal Resistance, 14L-PDIP	θ_{JA}	—	70	—	$^\circ C/W$	
Thermal Resistance, 14L-SOIC	θ_{JA}	—	120	—	$^\circ C/W$	
Thermal Resistance, 14L-TSSOP	θ_{JA}	—	100	—	$^\circ C/W$	

MCP601/602/603/604

2.0 TYPICAL PERFORMANCE CURVES

Note: Unless otherwise indicated, $V_{DD} = +2.7V$ to $+5.5V$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 25k\Omega$ to $V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$

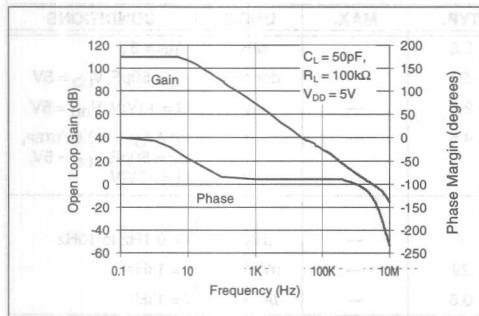


FIGURE 2-1: Open Loop Gain, Phase Margin vs. Frequency

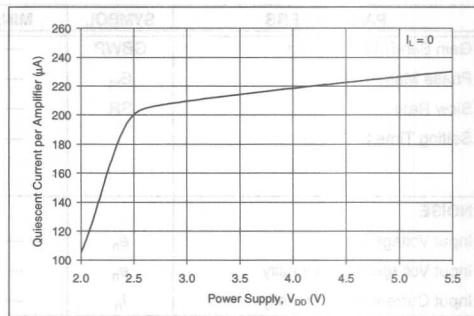


FIGURE 2-4: Quiescent Current vs. Power Supply

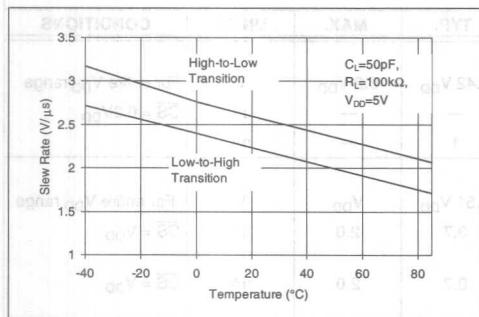


FIGURE 2-2: Slew Rate vs. Temperature

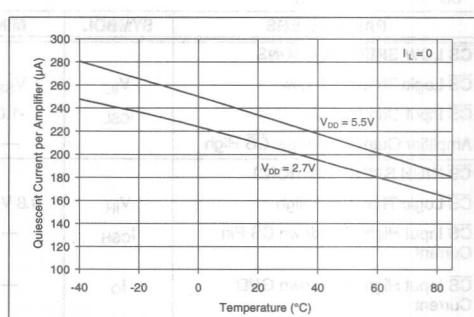


FIGURE 2-5: Quiescent Current vs. Temperature

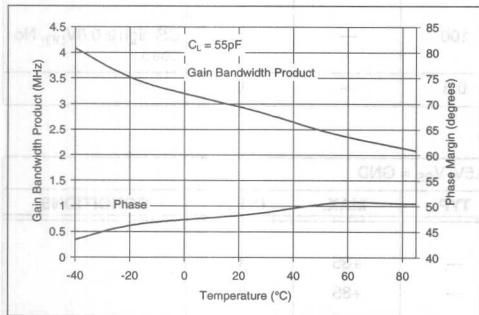


FIGURE 2-3: Gain Bandwidth Product vs. Temperature

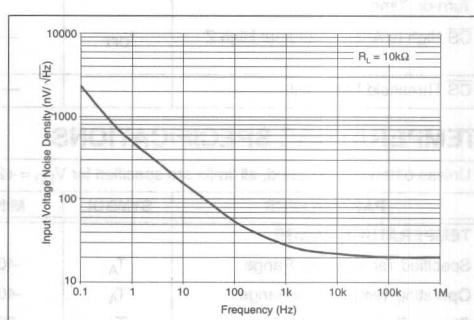


FIGURE 2-6: Input Voltage Noise Density vs. Frequency

MCP601/602/603/604

Note: Unless otherwise indicated, $V_{DD} = +2.7V$ to $+5.5V$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 25k\Omega$ to $V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$

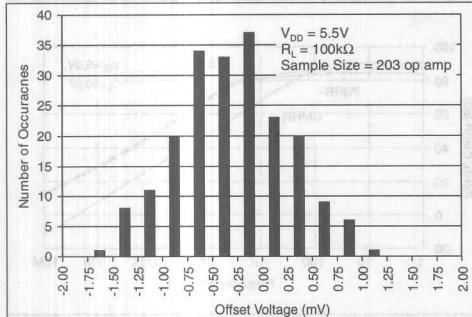


FIGURE 2-7: Offset Voltage vs. Number of Occurrences with $V_{DD} = 5.5V$

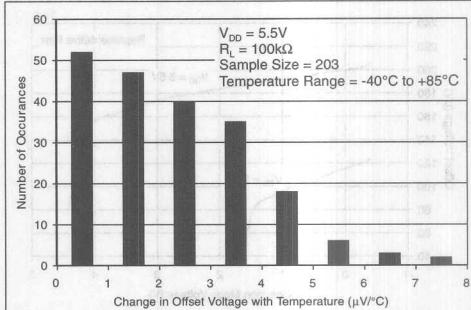


FIGURE 2-10: Offset Voltage Drift vs. Number of Occurrences with $V_{DD} = 5.5V$

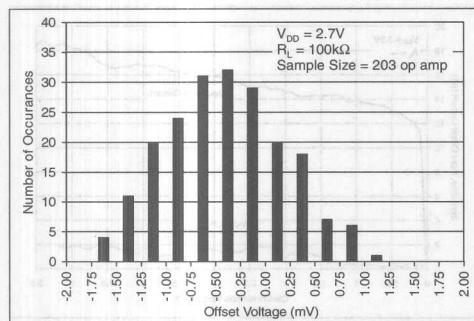


FIGURE 2-8: Offset Voltage vs. Number of Occurrences with $V_{DD} = 2.7V$

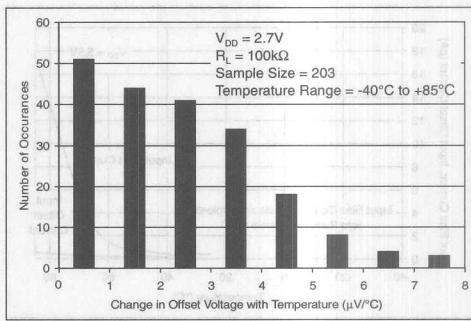


FIGURE 2-11: Offset Voltage Drift vs. Number of Occurrences with $V_{DD} = 2.7V$

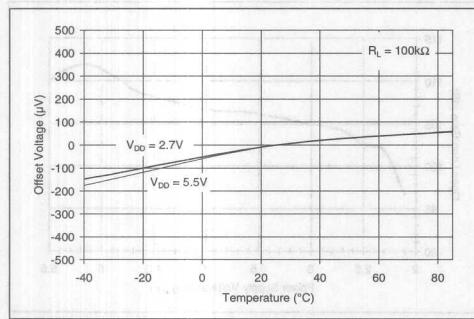


FIGURE 2-9: Normalized Offset Voltage vs. Temperature with $V_{DD} = 2.7V$

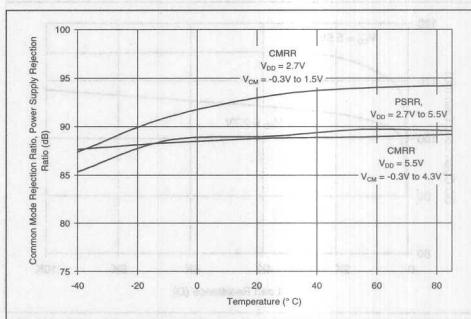


FIGURE 2-12: Common-Mode Rejection Ratio, Power Supply Rejection Ratio vs. Temperature

Note: Unless otherwise indicated, $V_{DD} = +2.7V$ to $+5.5V$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 25k\Omega$ to $V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$

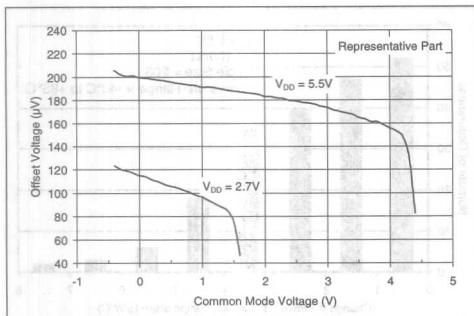


FIGURE 2-13: Offset Voltage vs. Common-Mode Voltage

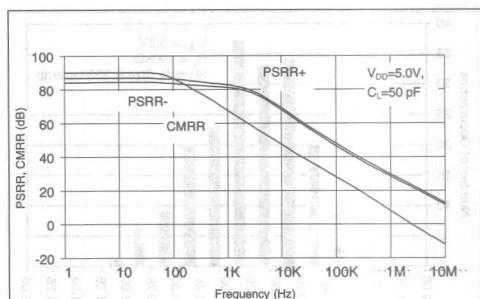


FIGURE 2-16: Common-Mode Rejection Ratio, Power Supply Rejection Ratio vs. Frequency

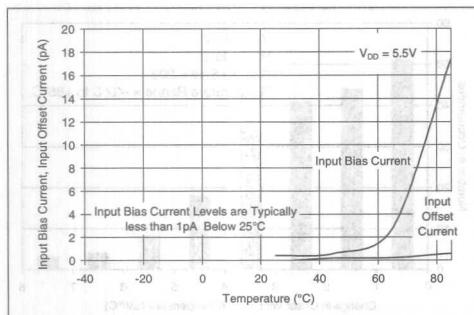


FIGURE 2-14: Input Bias Current, Input Offset Current vs. Temperature

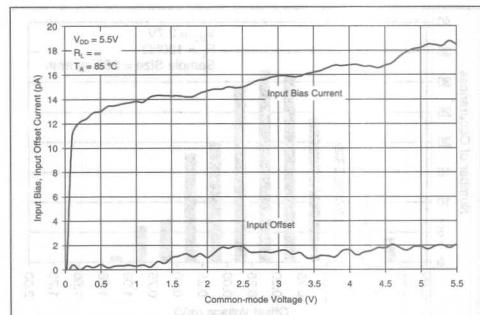


FIGURE 2-17: Input Bias Current, Input Offset Current vs. Common Mode Input Voltage

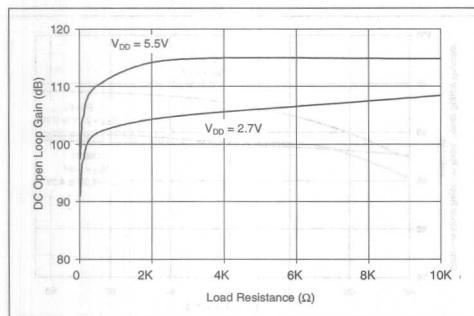


FIGURE 2-15: DC Open Loop Gain vs. Output Load

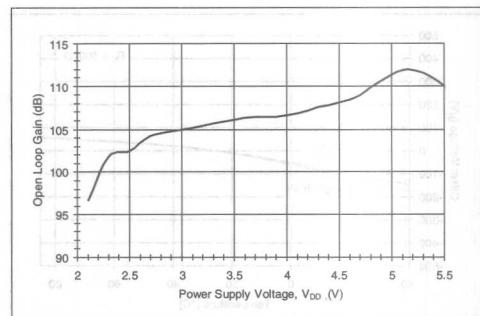


FIGURE 2-18: DC Open Loop Gain vs. Power Supply

MCP601/602/603/604

Note: Unless otherwise indicated, $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 25k\Omega$ to $V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$

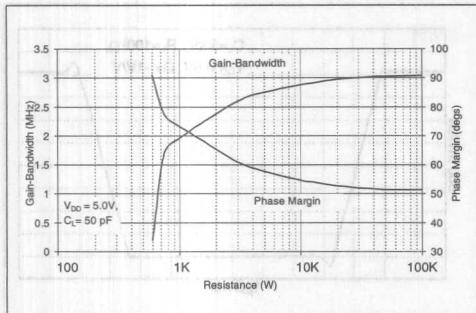


FIGURE 2-19: Gain Bandwidth, Phase Margin vs. Load Resistance

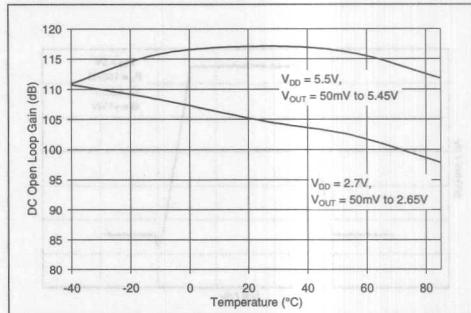


FIGURE 2-22: DC Open Loop Gain vs. Temperature

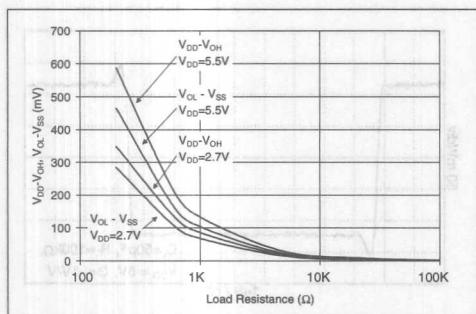


FIGURE 2-20: Low Level and High Level Output Swing vs. Resistive Load

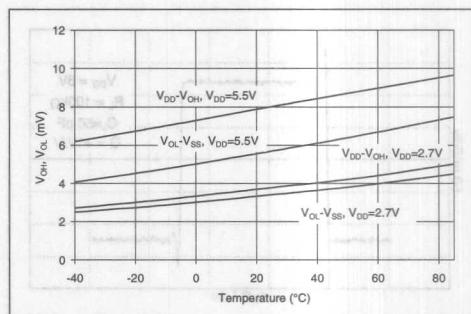


FIGURE 2-23: Low Level and High Level Output Swing vs. Temperature

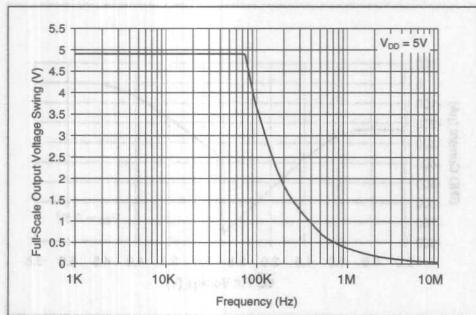


FIGURE 2-21: Maximum Full Scale Output Voltage Swing vs. Frequency

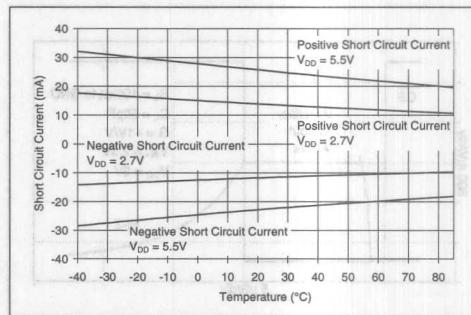


FIGURE 2-24: Output Short Circuit Current vs. Temperature

MCP601/602/603/604

Note: Unless otherwise indicated, $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 25k\Omega$ to $V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$

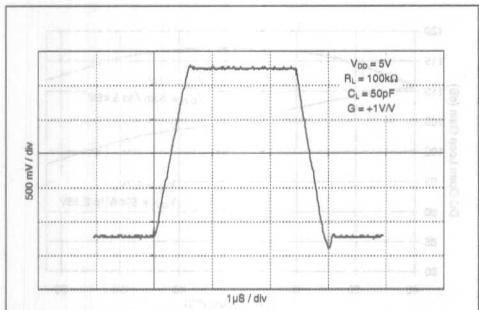


FIGURE 2-25: Large Signal Non-Inverting Signal Pulse Response

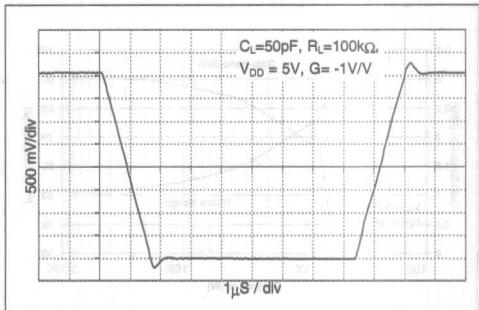


FIGURE 2-28: Large Signal Inverting Signal Pulse Response

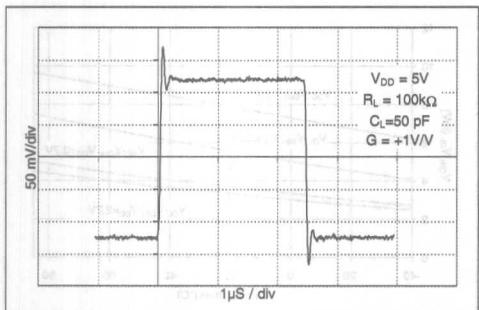


FIGURE 2-26: Small Signal Non-inverting Pulse Response

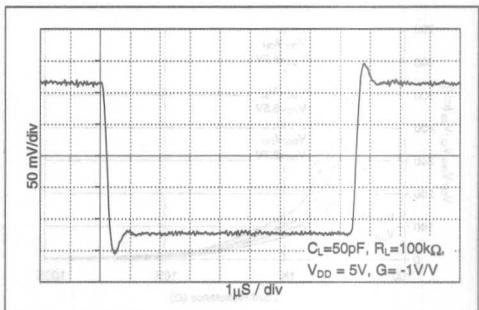


FIGURE 2-29: Small Signal Inverting Signal Pulse Response

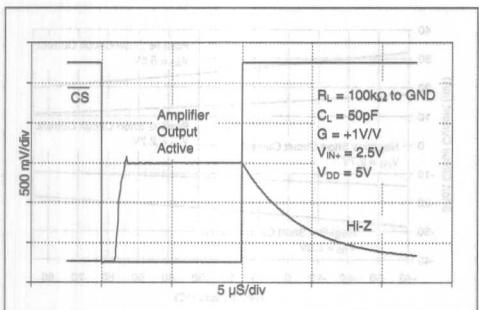


FIGURE 2-27: Chip Select to Amplifier Output Response Time

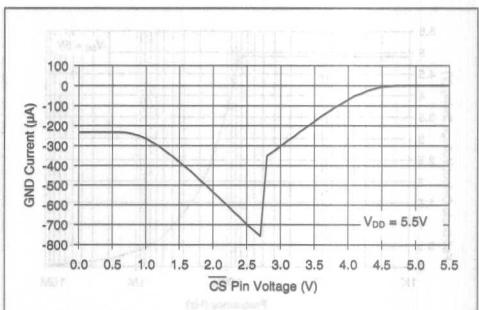


FIGURE 2-30: GND Current vs. CS Voltage

MCP601/602/603/604

Note: Unless otherwise indicated, $V_{DD} = +2.7V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^{\circ}C$, $V_{CM} = V_{DD}/2$, $R_L = 25k\Omega$ to $V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$

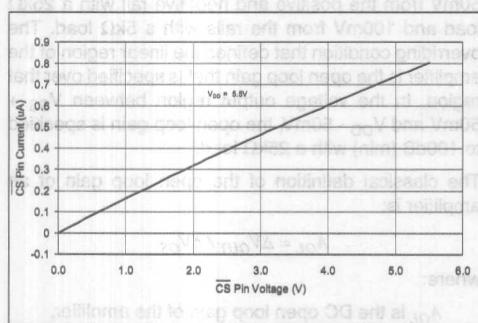


FIGURE 2-31: Input CS Current vs. CS Voltage

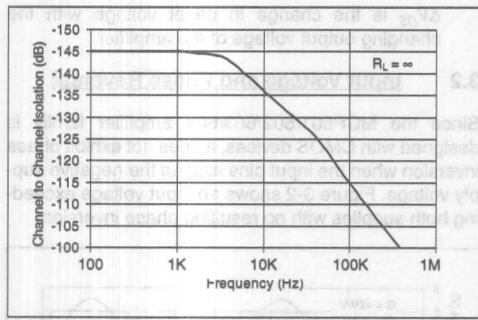


FIGURE 2-32: Channel to Channel Separation



FIGURE 2-33: CS hysteresis

The second section contains the electrical characteristics of the MCP601/602/603/604. These characteristics are defined over the operating range of V_{DD} and V_{SS} . The characteristics are divided into three main sections: DC characteristics, AC characteristics, and transient characteristics.

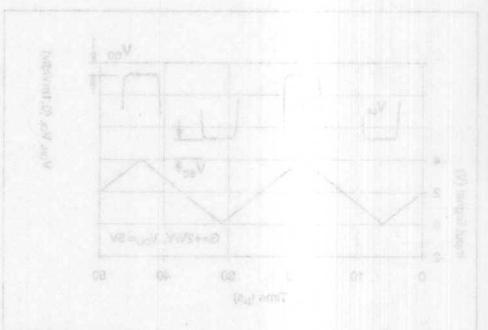


FIGURE 2-34: Turn On and Turn Off Transient Response

FIGURE 2-35: The MCP601/602/603/604 is a low noise, low distortion, high speed, high resolution, low power, monolithic operational amplifier. It features a unique input stage that provides high common mode rejection and low noise. The output stage is a current mirror, which provides high output current and low distortion. The MCP601/602/603/604 is ideal for applications such as audio processing, instrumentation, and control systems.

MCP601/602/603/604

3.0 APPLICATIONS INFORMATION

The MCP601/602/603/604 family of operational amplifiers are fabricated on Microchip's state-of-the-art CMOS process. They are unity gain stable and suitable for a wide range of general purpose applications. With this family of operational amplifiers, the power supply pin should be by-passed with a $1\mu\text{F}$ capacitor.

3.1 Rail-to-Rail Output Swing

There are two specifications that describe the output swing capability of the MCP601/602/603/604 family of operational amplifiers. The first specification, Low Level and High Level Output Voltage Swing, defines the absolute maximum swing that can be achieved under specified loaded conditions. For instance, the Low Level Output Voltage Swing of the MCP601/602/603/604 family is specified to be able to swing at least to 15mV from the negative rail with a $25\text{k}\Omega$ load to $V_{DD}/2$.

This output swing performance is shown in Figure 3-1, where the output of an MCP601 is configured in a gain of $+2\text{V/V}$ and over driven with a 40kHz triangle wave. In this figure, the degradation of the output swing linearity is clearly illustrated. This degradation occurs after the point at which the open loop gain of the amplifier is specified and before the amplifier reaches its maximum and minimum output swing.

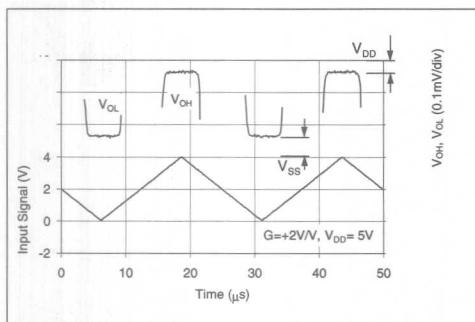


FIGURE 3-1: Low Level and High Level Output Swing

The second specification that describes the output swing capability of these amplifiers is the Linear Region Maximum Output Voltage Swing. This specification defines the maximum output swing that can be achieved while the amplifier is still operating in its linear region.

The Linear Region Maximum Output Voltage Swing of the MCP601/602/603/604 family is specified within 50mV from the positive and negative rail with a $25\text{k}\Omega$ load and 100mV from the rails with a $5\text{k}\Omega$ load. The overriding condition that defines the linear region of the amplifier is the open loop gain that is specified over that region. In the voltage output region between $V_{SS} + 50\text{mV}$ and $V_{DD} - 50\text{mV}$, the open loop gain is specified to 100dB (min) with a $25\text{k}\Omega$ load.

The classical definition of the open loop gain of an amplifier is:

$$A_{OL} = \Delta V_{OUT} / \Delta V_{OS}$$

where:

A_{OL} is the DC open loop gain of the amplifier,

ΔV_{OUT} is equal to $(V_{DD} - 50\text{mV}) - (V_{SS} + 50\text{mV})$ for $R_L = 25\text{k}\Omega$, and

ΔV_{OS} is the change in offset voltage with the changing output voltage of the amplifier.

3.2 Input Voltage and Phase Reversal

Since the MCP601/602/603/604 amplifier family is designed with CMOS devices, it does not exhibit phase inversion when the input pins exceed the negative supply voltage. Figure 3-2 shows an input voltage exceeding both supplies with no resulting phase inversion.

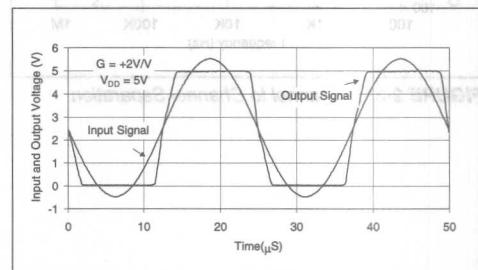


FIGURE 3-2: The MCP601/602/603/604 family of op amps do not have phase reversal issues. For the graph, the amplifier is in a unity gain or buffer configuration.

The maximum operating common-mode voltage that can be applied to the inputs is $V_{SS} - 0.3V$ to $V_{DD} + 1.2V$. In contrast, the absolute maximum input voltage is $V_{SS} - 0.3V$ and $V_{DD} + 0.3V$. Voltages on the input that exceed this absolute maximum rating can cause excessive current to flow in or out of the input pins. Current beyond $\pm 2mA$ can cause possible reliability problems. Applications that exceed this rating must be externally limited with an input resistor as shown in Figure 3-3.

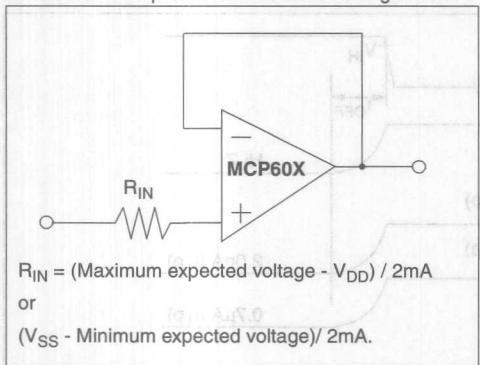


FIGURE 3-3: If the inputs of the amplifier exceed the Absolute Maximum Specifications, an input resistor, R_{IN} , should be used to limit the current flow into that pin.

3.3 Capacitive Load and Stability

Driving capacitive loads can cause stability problems with many of the higher speed amplifiers.

For any closed loop amplifier circuit, a good rule of thumb is to design for a phase margin that is no less than 45° . This is a conservative theoretical value, however, if the phase margin is lower, layout parasitics can degrade the phase margin further causing a truly unstable circuit. A system phase shift of 45° will have an overshoot in its step response of approximately 25%.

A buffer configuration with a capacitive load is the most difficult configuration for an amplifier to maintain stability. The Phase versus Capacitive Load of the MCP60X amplifier is shown in Figure 3-4. In this figure, it can be seen that the amplifier has a phase margin above 40° , while driving capacitance loads up to 100pF.

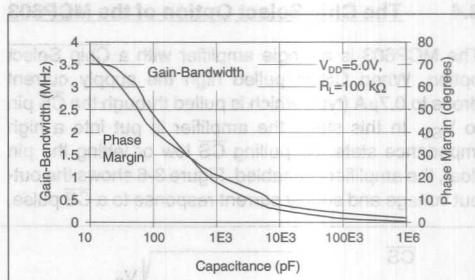


FIGURE 3-4: Gain Bandwidth, Phase Margin vs. Capacitive Load

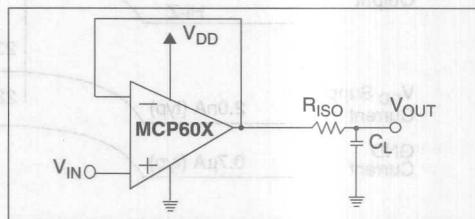


FIGURE 3-5: Amplifier circuits that can be used when driving heavy capacitive loads.

If the amplifier is required to drive larger capacitive loads, the circuit shown in Figure 3-5 can be used. A small series resistor (R_{ISO}) at the output of the amplifier improves the phase margin when driving large capacitive loads. This resistor decouples the capacitive load from the amplifier by introducing a zero in the transfer function.

This zero adjusts the phase margin by approximately:

$$\Delta\theta_m = \tan^{-1}(2\pi GBWP \times R_{ISO} \times C_L)$$

where:
 $\Delta\theta_m$ is the improvement in phase margin,
 $GBWP$ is the gain bandwidth product of the amplifier,

R_{ISO} is the capacitive decoupling resistor, and
 C_L is the load capacitance

MCP601/602/603/604

3.4 The Chip Select Option of the MCP603

The MCP603 is a single amplifier with a Chip Select option. When \overline{CS} is pulled high the supply current drops to $0.7\mu A$ (typ), which is pulled through the \overline{CS} pin to V_{SS} . In this state, the amplifier is put into a high impedance state. By pulling \overline{CS} low or letting the pin float, the amplifier is enabled. Figure 3-6 shows the output voltage and supply current response to a \overline{CS} pulse.

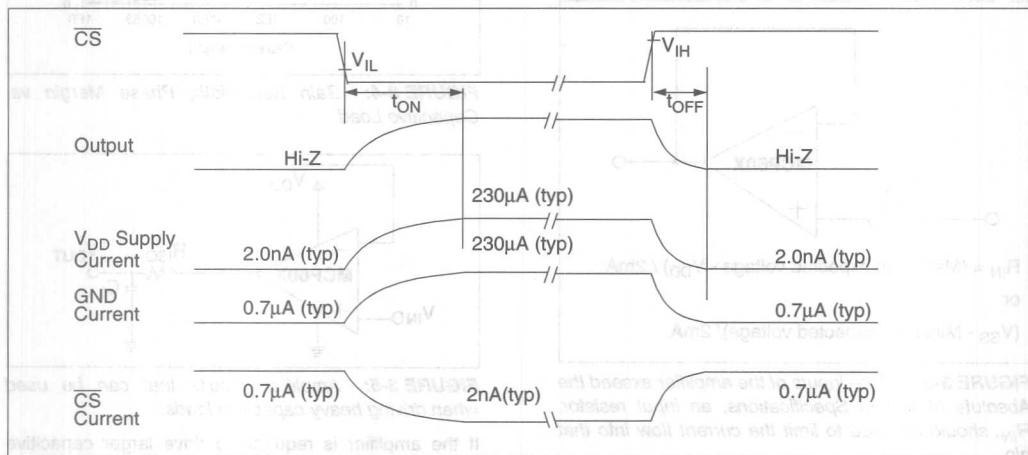


FIGURE 3-6: Timing Diagram for the \overline{CS} Function of the MCP603 Amplifier

3.5 Layout Considerations

In applications where low input bias current is critical, PC board surface leakage effects and signal coupling from trace to trace need to be taken into consideration.

3.5.1 SURFACE LEAKAGE

Surface leakage across a PC board is a consequence of differing DC voltages between two traces combined with high humidity, dust or contamination on the board. For instance, the typical resistance from PC board trace to pad is approximately $10^{12}\Omega$ under low humidity conditions. If an adjacent trace is biased to 5V and the input pin of the amplifier is biased at or near zero volts, a 5pA leakage current will appear on the amplifier's input node. This type of PCB leakage is five times the room temperature input bias current (1pA, typ) of the MCP601/602/603/604 family of amplifiers.

The simplest technique that can be used to reduce the effects of PC board leakage is to design a ring around sensitive pins and traces. An example of this type of layout is shown in Figure 3-7.

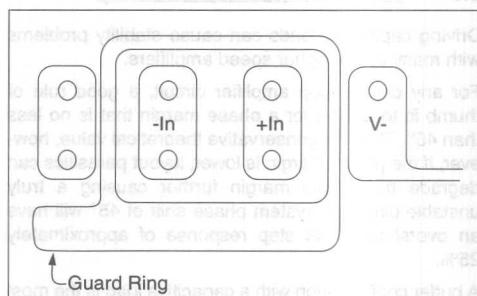


FIGURE 3-7: Example of Guard Ring for the MCP601, the A-amplifier of the MCP602 or the MCP603 in a PC Board Layout

Circuit examples of ring implementations are shown in Figure 3-8. In Figure 3-8A, B and C, the guard ring is biased to the common-mode voltage of the amplifier. This type of guard ring is most effective for applications where the common-mode voltage of the input stage changes, such as buffers, inverting gain amplifiers or instrumentation amplifiers.

The strategy shown in Figure 3-8D, biases the common-mode voltage and guard ring to ground. This type of guard ring is typically used in precision photo sensing circuits.

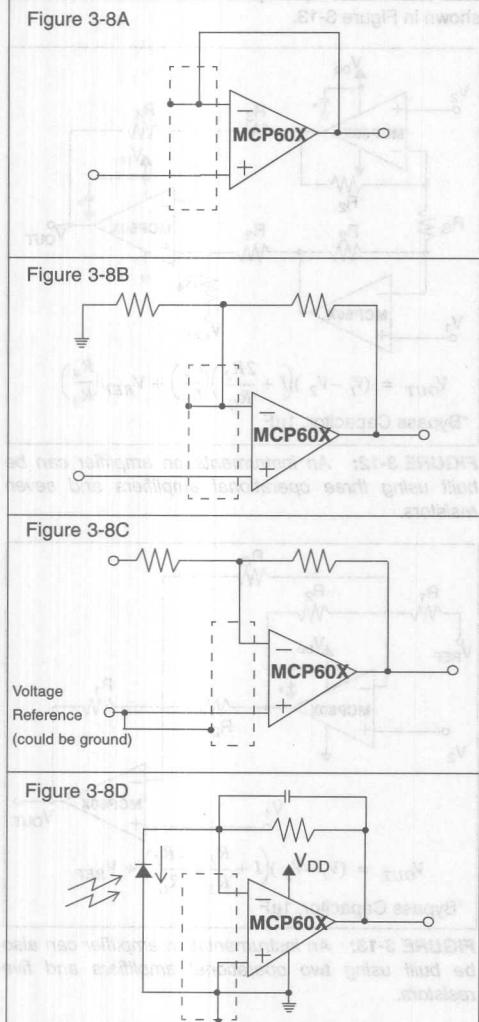


FIGURE 3-8: Examples of how to design PC Board traces to minimize leakage paths to the high impedance input pins of the MCP601/602/603/604 amplifiers.

3.5.2 SIGNAL COUPLING

The input pins of the MCP601/602/603/604 amplifiers have a high impedance providing an opportunity for noise injection, if layout issues are not considered. These high impedance input terminals are sensitive to injected currents. This can occur if the trace from a high impedance input is next to a trace that has fast changing voltages, such as a digital or clock signal. When a high impedance trace is in close proximity to a trace with these types of voltage changes, charge is capacitively coupled into the high impedance trace.

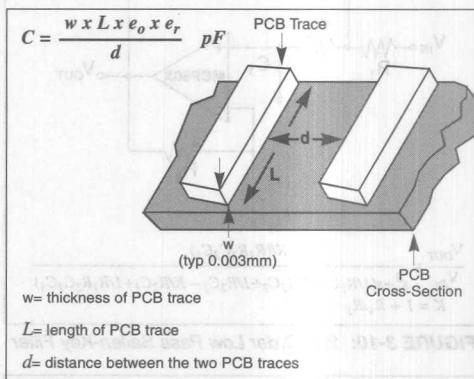


FIGURE 3-9: Capacitors can be built with PCB traces allowing for coupling of signals from one trace to another.

As shown in Figure 3-9, the value of the capacitance between two traces is primarily dependent on the distance (d) between the traces and the distance that the two traces are in parallel (L). From this model, the amount of current generated into the high impedance trace is equal to:

$$I = C \frac{\partial V}{\partial t}$$

where:

I equals the current that appears on the high impedance trace,

C equals the value of capacitance between the two PCB traces,

∂V equals the change in voltage of the trace that is switching, and

∂t equals the amount of time that the voltage change took to get from one level to the next.

MCP601/602/603/604

3.6 Typical Applications

3.6.1 ANALOG FILTERS for applications such as audio, communications, medical, industrial, and instrumentation.

Examples of two second order low pass filters are shown in Figure 3-10 and Figure 3-11. The filter in Figure 3-10 can be configured for gain of +1V/V or greater. The filter in Figure 3-11 can be configured for inverting gains.

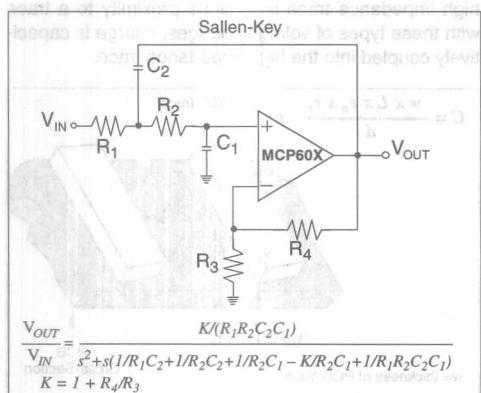


FIGURE 3-10: 2nd Order Low Pass Sallen-Key Filter

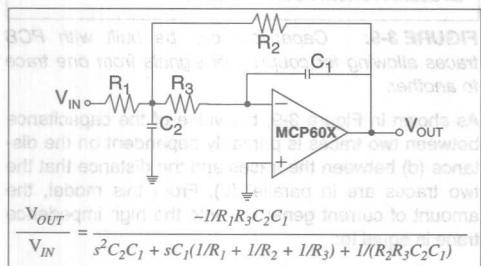


FIGURE 3-11: 2nd Order Low Pass Multiple-Feedback Filter

The MCP601/602/603/604 family of operational amplifiers are particularly well suited for these types of filters. The low input bias current, which is typically 1pA (up to 60pA at temperature), allows the designer to select higher value resistors, which in turn reduces the capacitive values. This allows the designer to select surface mount capacitors, which in turn can produce a compact layout.

The rail-to-rail output operation of the MCP601/602/603/604 family of amplifiers make these circuits well suited for single supply operation. Additionally, the wide bandwidth allows low pass filter design up to 1/10 of the GBWP or 300kHz.

These filters can be designed using the calculations provided in the Figures or with Microchip's interactive **FilterLab** software. **FilterLab** will calculate capacitor and resistor values, as well as, determine the number

of poles that are required for the application. Finally, the program will generate a SPICE macromodel, which can be used for spice simulations.

3.6.2 INSTRUMENTATION AMPLIFIER CIRCUITS

The instrumentation amplifier has a differential input, which subtracts one analog signal from another and rejects common mode signals. This amplifier also provides a single ended analog output signal. The three op amp instrumentation amplifier is illustrated in Figure 3-12 and the two op amp instrumentation amplifier is shown in Figure 3-13.

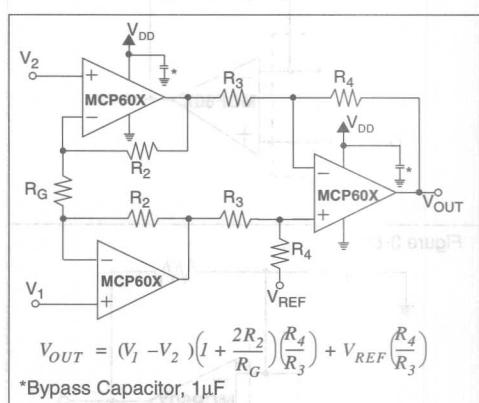


FIGURE 3-12: An instrumentation amplifier can be built using three operational amplifiers and seven resistors.

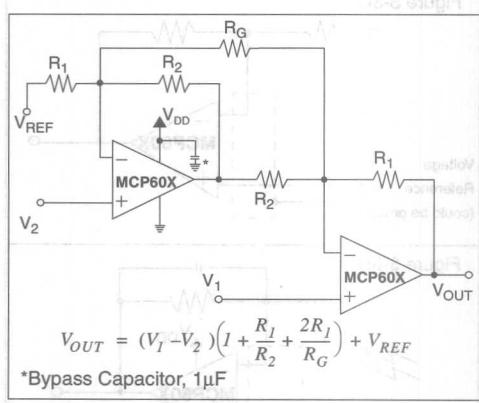


FIGURE 3-13: An instrumentation amplifier can also be built using two operational amplifiers and five resistors.

An advantage of the three op amp configuration is that it is capable of unity gain operation. A disadvantage, as compared to the two op amp instrumentation amplifier, is that the common mode range reduces with higher gains.

The two op amp configuration uses fewer op amps, so power consumption is also low. Disadvantages of this configuration are that the common-mode range reduces with gain and it must be configured in gains of two or higher.

3.6.3 PHOTO DETECTION

The amplifiers in the MCP601/602/603/604 family of devices can be used to easily convert the signal from a sensor that produces an output current, such as a photodiode, into a voltage. This is implemented with a single resistor and an optional capacitor in the feedback loop of the amplifier as shown in Figure 3-14.

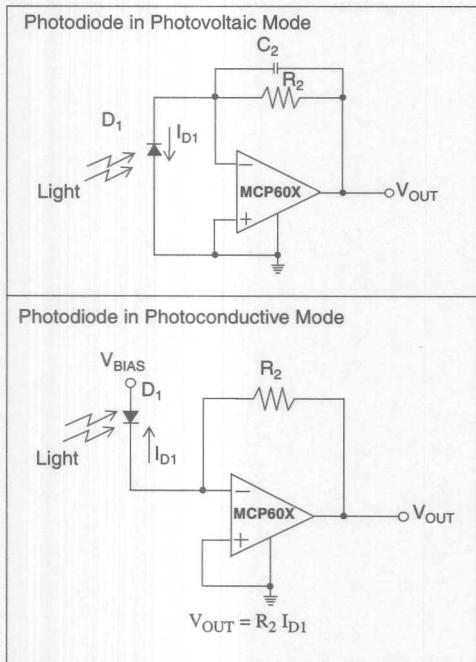


FIGURE 3-14: Photo Sensing Circuits Using the MCP60X Amplifier

A photodiode that is configured in the photovoltaic mode has no voltage potential placed across the element or is zero biased (Figure 3-14). In this mode, the light sensitivity and linearity is maximized making it best suited for precision applications. The key amplifier specifications for this application are low input bias current, low noise and rail-to-rail output swing. The MCP601/602/603/604 family is capable of meeting all three of these difficult requirements.

In contrast, a photodiode that is configured in the photoconductive mode has a reverse bias voltage, which is applied across the photo sensing element as shown in Figure 3-14. The width of the depletion region is reduced when this voltage is applied across the photo detector, which reduces the photodiode parasitic capacitance significantly. This reduced parasitic capacitance facilitates high speed operation, however, the linearity and offset errors are not optimized. The design trade off for this action is increased diode leakage current and linearity errors. A key amplifier specification for this application is high speed digital communication. The MCP601/602/603/604 family is well suited for medium speed photoconductive applications with their wide bandwidth and rail-to-rail output swing.

MCP601/602/603/604

4.0 SPICE MACROMODEL

The Spice macromodel for the MCP601, MCP602, MCP603 and MCP604 simulates the typical amplifier performance of offset voltage, DC power supply rejection, input capacitance, DC common mode rejection ratio, open loop gain over frequency, phase margin with no capacitive load, output swing, DC power supply current, power supply current change with supply voltage, input common mode range and input voltage noise.

The characteristics of the MCP601, MCP602, MCP603, and MCP604 amplifiers are similar in terms of performance and behavior. This single op amp macromodel supports all four devices with the exception of the chip select function of the MCP603, which is not modeled.

The listing for this macromodel is shown on the next page. The most recent revision of the model can be downloaded from Microchip's web site at www.microchip.com.

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the "Company") for its PICmicro® Microcontroller is intended and supplied to you, the Company's customer, for use solely and exclusively on Microchip PICmicro Microcontroller products.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

```
.subckt mcp601 1 2 3 4 5
*           | | | |
*           | | | Output
*           | | | Negative supply
*           | | Positive Supply
*           | Inverting input
*           Non-inverting input
*
* Macromodel for MCP601 (single), MCP602 (dual), MCP603 (single w/CS), and MCP604 (quad)
*
* The characteristics of the MCP601, MCP602, MCP603, and MCP604 have the same fundamental
* performance and behavior. Consequently, this single op amp macromodel supports all four
* devices. However, the chip select function of the MCP603 is not modeled.
*
* Revision History:
*   REV A : 6-30-99 created BCB
*   REV B : 7-10-99 corrected DC Iq BCB
*   REV C : 11-30-99 Placed ".subckt" command as first line, added L, W to Ptype model in
*           : listing BCB
*
* This macromodel models typical amplifier offset voltage, DC power supply rejection, input
* capacitance, DC common mode rejection ratio, open loop gain over frequency, phase margin
* with no capacitive load, output swing, power supply current, input voltage noise.
*
* NOTICE: THE INFORMATION PROVIDED HEREIN IS BELIEVED TO BE RELIABLE,
* HOWEVER, MICROCHIP ASSUMES NO RESPONSIBILITY FOR INACCURACIES OR
* OMISSIONS. MICROCHIP ASSUMES NO RESPONSIBILITY FOR THE USE OF THIS
* INFORMATION, AND ALL USE OF SUCH INFORMATION SHALL BE ENTIRELY AT
* THE USER'S OWN RISK. NO INTELLECTUAL PROPERTY RIGHTS OR LICENSES
* TO ANY OF THE TECHNOLOGY DESCRIBED HEREIN ARE IMPLIED OR GRANTED TO
* ANY THIRD PARTY. MICROCHIP RESERVES THE RIGHT TO CHANGE THIS MODEL
* AT ANY TIME WITHOUT NOTICE.
*
*Input Stage, pole at 5MHz
M1    9      64     7      3      Ptype L=2 W=275
M2    8      2      7      3      Ptype L=2 W=275
CDIFF  1      2      3E-12
CCM1   1      4      6E-12
CCM2   2      4      6E-12
IDD    3      7      30e-6
RA     8      6      1.485e3
RB     9      6      1.485e3
CA     8      9      10.71e-12
*
*Input Stage Common-Mode Clamping
VCMC   4      6      0.35
ECM    55     4      3 64    1
RCM    57     56     1E3
DCMP   56     55     DX
VCMP   57     4      1.2
```

MCP601/602/603/604

```

RST    58   59   1E3
DST    59   55   DX
VST    58   4    1.6

GCMP2  23   4    POLY(2)  57 56 58 59 0 -0.5E-3 0.5E-3

*Input errors (vos, en, psr, cmr)
ERR    64   1    POLY(3)  (67,4) (3,4) (1,34) 0 1 40e-6 3.2e-6

*Second Stage, pole at 3.3Hz
GS     23   4    8      9      5.7e-3
R1     23   4    0.397e9
C2     23   4    122.8e-12

VSOP    3    24   4.784
VSOM   25    4   -3.48
DSOP   23    24   DY
DSOM   25    23   DY

*HCM    23    3    VCMP

FS 3 4 POLY(11) V03 V05 V04 V06 V01 V02 V09 V010 VMID1 VSOP VSOM
+ 200E-6 -1 -1 -1 1 -1 -1 1 1 -1 -1 -1

*mid-supply reference, output swing limit
RMID1  3    35   61.62E3
VMID1  35   34   0
RMID2  4    34   61.62E3
ELEVEL 34   4    23      4      -1

*DOutput stage
D03    34   43   DY
D04    44   34   DY
D05    3    45   DY
D06    3    46   DY
D07    4    45   DY
D08    4    46   DY
V03    43   5    0.1
V04    5    44   0.1
G05    3    47   3    34  10E-3
V05    47   5    0
G06    4    48   34   4  10E-3
V06    48   5    0
G01    49   4    5    34  10E-3
V01    49   45   0
G02    50   4    34   5    10E-3
V02    50   46   0
R09    3    51   100
V09    51   5    0
R010   52   4    100
V010   52   5    0

* input voltage noise
VN1    65   4    0.6
DN1    65   67   DX
RN1    67   4    13E3

.model Ptype PMOS
.model DY D(IS=1e-15 BV =50)
.model DX D(IS=1e-18 AF=0.6 KF=10e-17)
.ENDS

```

MCP601/602/603/604

MCP60X PRODUCT IDENTIFICATION SYSTEM

:SETON

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

MCP60X — X /X	Package:	P = Plastic DIP (300 mil Body), 8-lead and 14-lead	
		SN = Plastic SOIC (150 mil Body), 8-lead	
	Temperature Range:	SL = Plastic SOIC (150 mil Body), 14-lead	
		ST = Plastic TSSOP, 8-lead and 14-lead	
		OT = Plastic SOT23, 5-lead	
	Device:	I = -40°C to +85°C	
		 MCP601 = Single Operational Amplifier MCP601T = Single Operational Amplifier (Tape and Reel-SOIC/TSSOP/SOT23-5) MCP602 = Dual Operational Amplifier MCP602T = Dual Operational Amplifier (Tape and Reel-SOIC/TSSOP) MCP603 = Single Operational Amplifier w/ \overline{CS} Function MCP603T = Single Operational Amplifier w/ \overline{CS} Function (Tape and Reel-SOIC/TSSOP) MCP604 = Quad Operational Amplifier MCP604T = Quad Operational Amplifier (Tape and Reel-SOIC/TSSOP)	

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

3

Datasheets

MCP60X

NOTES:



MICROCHIP

MCP606/607/608/609

2.5V to 5.5V Micropower CMOS Op Amps

FEATURES

- Low Power $I_{DD} = 25\mu A$, max
- Low Offset Voltage: $250\mu V$, max
- Rail-to-Rail Swing at Output
- $80pA$, Low Input Bias Current over Temperature
- Specifications rated for 2.5V to 5.5V Supplies
- Unity Gain Stable
- Chip Select Capability with MCP608
- Industrial Temperature range supported
- No Phase Reversal
- Available in Single, Dual, and Quad

APPLICATIONS

- Battery Power Instruments
- High Impedance Applications
 - Photodiode Pre-amps
 - pH probe Buffer Amplifier
 - Infrared Detectors
 - Precision Integrators
 - Charge Amplifier for Piezoelectric Transducers
- Strain Gauges
- Medical Instruments
- Test Equipment

AVAILABLE TOOLS

- Spice Macromodels (at www.microchip.com)
- **FilterLab™ Software** (at www.microchip.com)

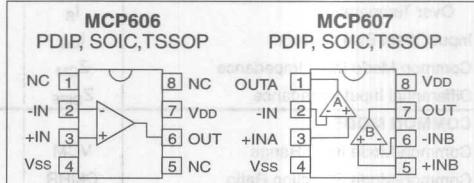
© 2000 Microchip Technology Inc.

DESCRIPTION

The MCP606, MCP607, MCP608 and MCP609 from Microchip Technology, Inc. are unity gain stable, low offset voltage operational amplifiers capable of precision low power single supply operation. Performance characteristics include ultra low offset voltage ($250\mu V$, max), rail-to-rail output swing capability, and low input bias current ($80pA$ @ $85^{\circ}C$). These features make this family of amplifiers well suited for single supply precision, high impedance, battery powered applications.

The single MCP606 is available in standard 8-lead PDIP, SOIC, and TSSOP packages. Another version of the single op amp, MCP608 is offered with a Chip Select option in standard 8-lead PDIP, SOIC, and TSSOP packages. The dual MCP607 is offered in standard 8-lead PDIP, SOIC, as well as the TSSOP package. Finally, the quad MCP609 is offered in 14-lead PDIP, SOIC and TSSOP packages. All devices are fully specified from $-40^{\circ}C$ to $+85^{\circ}C$ with power supplies from 2.5V to 5.5V.

PACKAGES



3

Datasheets

MCP606/607/608/609

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.3V to $V_{DD} + 0.3V$
Difference Input voltage	$ V_{DD} - V_{SS} $
Output Short Circuit Current	continuous
Current at Input Pin	$\pm 2\text{mA}$
Current at Output and Supply Pins	$\pm 30\text{mA}$
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-55°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	$\geq 2\text{kV}$

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the data sheet is not implied. Functional operation is limited by design and process specifications only and does not imply that the device will function under all conditions above those indicated.

operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
+IN/+INA/+INB/+INC/+IND	Non-inverting Input Terminals
-IN/-INA/-INB/-INC/-IND	Inverting Input Terminals
V_{DD}	Positive Power Supply
V_{SS}	Negative Power Supply
OUT/OUTA/OUTB/OUTC/OUTD	Output Terminals
C_S	Chip Select
NC	No internal connection to IC

DC CHARACTERISTICS

Unless otherwise specified, all limits are specified for $V_{DD} = +2.5\text{V}$ to $+5.5\text{V}$, $V_{SS} = \text{GND}$, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$, $R_L = 100\text{k}\Omega$ to $V_{DD}/2$, and $V_{OUT} \sim V_{DD}/2$.

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
INPUT OFFSET						
Input Offset Voltage	V_{OS}	-250	—	+250	μV	
Drift with Temperature,	dV_{OS}/dT	—	± 1.8	—	$\mu\text{V}/^\circ\text{C}$	$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$
Power Supply Rejection	PSRR	80	93	—	dB	for $V_{DD} = 2.5\text{V}$ to 5.5V
INPUT BIAS CURRENT AND IMPEDANCE						
Input Bias Current	I_B	—	1	—	pA	
Over Temperature	I_B	—	—	80	pA	$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$
Input Offset Bias Current	I_{OS}	—	1	—	pA	
Common Mode Input Impedance	Z_{CM}	—	10^{13}Il6	—	ΩIlpF	
Differential Input Impedance	Z_{DIFF}	—	10^{13}Il6	—	ΩIlpF	
COMMON MODE						
Common-Mode Input Range	V_{CM}	$V_{SS} - 0.3$	91	$V_{DD} - 1.1$	V	$\text{CMRR} > 75\text{dB}$
Common-Mode Rejection Ratio	CMRR	75	—	—	dB	$V_{DD} = 5\text{V}$, $V_{CM} = -0.3$ to 3.9V
OPEN LOOP GAIN						
DC Open Loop Gain	A_{OL}	105	121	—	dB	$R_L = 25\text{k}\Omega$ to GND, $50\text{mV} < V_{OUT} < (V_{DD} - 50\text{mV})$
DC Open Loop Gain	A_{OL}	100	118	—	dB	$R_L = 5\text{k}\Omega$ to GND, $100\text{mV} < V_{OUT} < (V_{DD} - 100\text{mV})$
OUTPUT						
Low Level/High Level Output Voltage Swing	V_{OL}, V_{OH}	$V_{SS} + 0.015$	—	$V_{DD} - 0.020$	V	$R_L = 25\text{k}\Omega$ to $V_{DD}/2$
	V_{OL}, V_{OH}	$V_{SS} + 0.045$	—	$V_{DD} - 0.060$	V	$R_L = 5\text{k}\Omega$ to $V_{DD}/2$
Linear Region Maximum Output Voltage Swing	V_{OUT}	$V_{SS} + 0.050$	—	$V_{DD} - 0.050$	V	$R_L = 25\text{k}\Omega$ to $V_{DD}/2$, $A_{OL} \geq 105\text{dB}$
	V_{OUT}	$V_{SS} + 0.100$	—	$V_{DD} - 0.100$	V	$R_L = 5\text{k}\Omega$ to $V_{DD}/2$, $A_{OL} \geq 100\text{dB}$
Output Short Circuit Current	I_{SC}	—	17	—	mA	$V_{OUT} = 2.5\text{V}$, $V_{DD} = 5\text{V}$
POWER SUPPLY						
Supply Voltage	V_S	2.5	—	5.5	V	
Quiescent Current Per Amp	I_Q	—	18.7	25	μA	$I_Q = 0$

AC CHARACTERISTICS

Unless otherwise specified, all limits are specified for $V_{DD} = +2.5V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 100k\Omega$ to $V_{DD}/2$, and $V_{OUT} \sim V_{DD}/2$

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Gain Bandwidth Product	GBWP	—	155	—	kHz	$V_{DD} = 5V$, $C_L = 60 pF$
Phase Margin at Unity Crossing	θ	—	62	—	degrees	$V_{DD} = 5V$, $C_L = 60 pF$
Slew Rate	SR	—	0.08	—	V/ μ s	$G = 1$, $V_{DD} = 5V$, $C_L = 60 pF$
Input Voltage Noise	e_n	—	2.8	—	μV_{p-p}	$f = 0.1Hz$ to $10Hz$
Noise Density	e_n	—	38	—	nV/ \sqrt{Hz}	$f = 1kHz$
Input Current Noise Density	i_n	—	3	—	fA/ \sqrt{Hz}	$f = 1kHz$

SPECIFICATIONS FOR MCP608 CHIP SELECT FEATURE

Unless otherwise specified, all limits are specified for $V_{DD} = +2.5V$ to $+5.5V$, $V_{SS} = GND$, $T_A = 25^\circ C$, $V_{CM} = V_{DD}/2$, $R_L = 100k\Omega$, and $V_{OUT} \sim V_{DD}/2$

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
CS LOW SPECIFICATIONS						
CS Logic Threshold, Low	V_{IL}	V_{SS}	—	0.2 V_{DD}	V	For entire V_{DD} range
CS Input Current, Low	I_{CSL}	-0.1	0.01	—	μA	$CS = 0.2V_{DD}$
CS HIGH SPECIFICATIONS						
CS Logic Threshold, High	V_{IH}	0.8 V_{DD}	—	V_{DD}	V	For entire V_{DD} range
CS Input Current, High	I_{CSH}	—	0.01	0.1	μA	$CS = V_{DD}$
CS Input High, GND Current	I_Q	—	0.05	0.1	μA	$CS = V_{DD}$
Amplifier Output Leakage, CS High	—	—	10	—	nA	$CS = 0.8V_{DD}$
DYNAMIC SPECIFICATIONS						
CS Low to Amplifier Output High Turn-on Time	t_{ON}	—	9	100	μs	CS low = $0.2V_{DD}$, $V_{OUT} = 0.9 * V_{DD}/2$, $G = +1V/V$
CS High to Amplifier Output High Z Hysteresis	t_{OFF}	—	0.1	—	μs	CS high = $0.8V_{DD}$, $V_{OUT} = 0.1 * V_{DD}/2$, $G = +1V/V$
	—	—	0.6	—	V	$V_{DD} = 5V$

TEMPERATURE SPECIFICATIONS

Unless otherwise specified, all limits are specified for $V_{DD} = +2.5V$ to $+5.5V$, $V_{SS} = GND$

PARAMETERS	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TEMPERATURE RANGES						
Specified Temperature Range	T_A	-40	—	+85	°C	
Operating Temperature Range	T_A	-40	—	+85	°C	
Storage Temperature Range	T_A	-65	—	+150	°C	
 THERMAL PACKAGE RESISTANCE						
Thermal Resistance, 8L-PDIP	θ_{JA}	—	85	—	°C/W	
Thermal Resistance, 8L-SOIC	θ_{JA}	—	163	—	°C/W	
Thermal Resistance, 8L-TSSOP	θ_{JA}	—	124	—	°C/W	
Thermal Resistance, 14L-PDIP	θ_{JA}	—	70	—	°C/W	
Thermal Resistance, 14L-SOIC	θ_{JA}	—	120	—	°C/W	
Thermal Resistance, 14L-TSSOP	θ_{JA}	—	100	—	°C/W	

MCP606/607/608/609

2.0 TYPICAL PERFORMANCE CURVES

Note: Unless otherwise indicated, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$, $V_{SS} = \text{GND}$

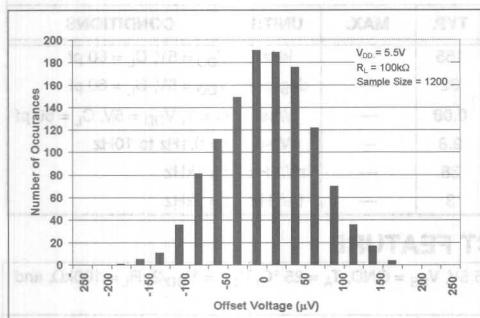


FIGURE 2-1: Offset Voltage vs. Number of Occurrences with $V_{DD} = 5.5\text{V}$

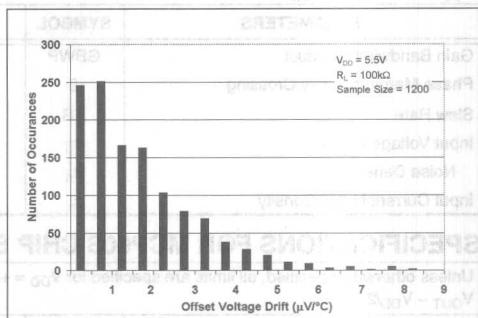


FIGURE 2-4: Offset Voltage Drift vs. Number of Occurrences with $V_{DD} = 5.5\text{V}$

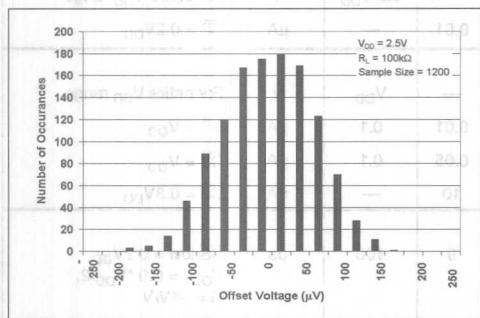


FIGURE 2-2: Offset Voltage vs. Number of Occurrences with $V_{DD} = 2.5\text{V}$

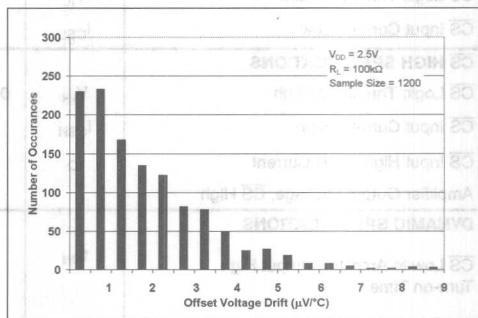


FIGURE 2-5: Offset Voltage Drift vs. Number of Occurrences with $V_{DD} = 2.5\text{V}$

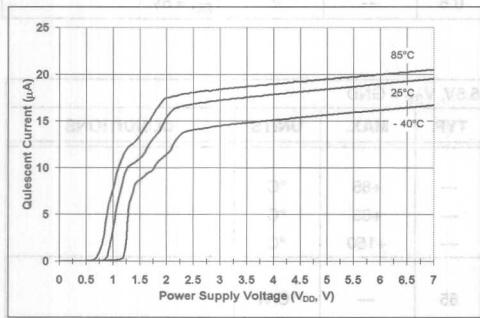


FIGURE 2-3: Quiescent Current vs. Power Supply Voltage vs. Temperature

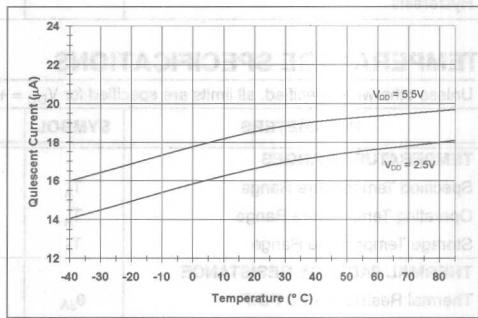


FIGURE 2-6: Quiescent Current vs. Temperature

Note: Unless otherwise indicated, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$, $V_{SS} = \text{GND}$ unless otherwise specified.

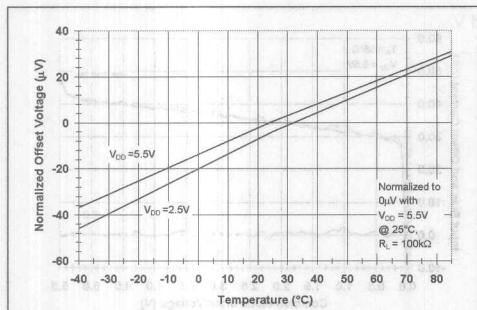


FIGURE 2-7: Normalized Offset Voltage vs. Temperature

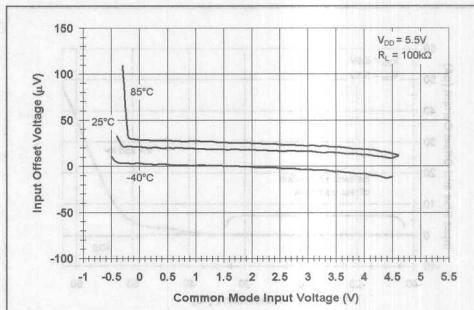


FIGURE 2-10: Input Offset Voltage vs. Common Mode Input Voltage

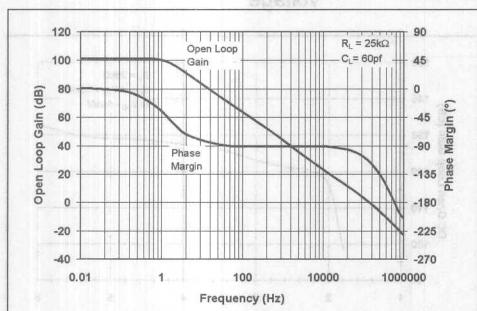


FIGURE 2-8: Open Loop Gain, Phase Margin vs. Frequency

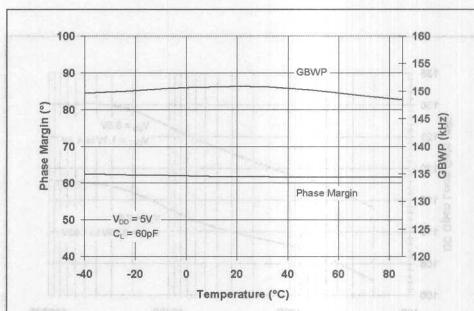


FIGURE 2-11: Phase Margin, Gain Bandwidth Product vs. Temperature

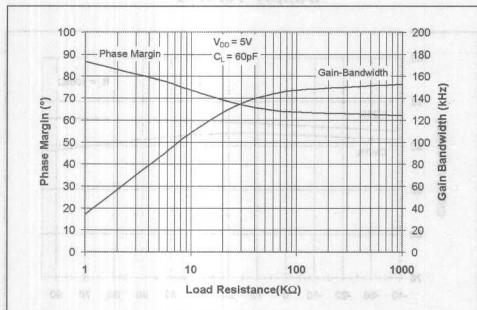


FIGURE 2-9: Phase Margin, Gain Bandwidth, vs. Load Resistance

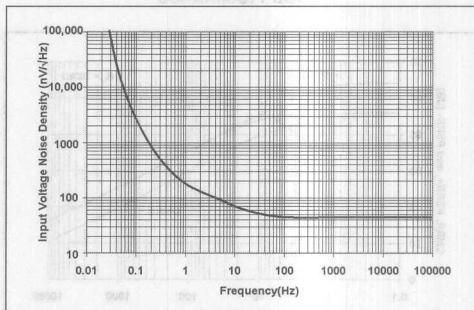


FIGURE 2-12: Input Voltage Noise Density vs. Frequency

MCP606/607/608/609

Note: Unless otherwise indicated, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$, $V_{SS} = \text{GND}$ if shown as ground connection.

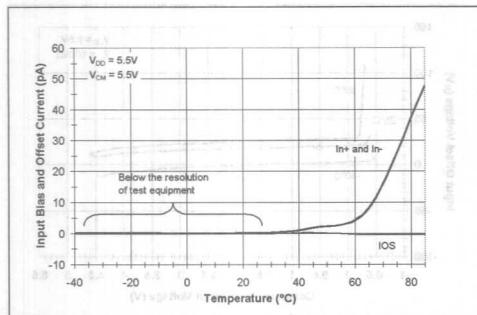


FIGURE 2-13: Input Bias Current, Input Offset Current vs. Temperature

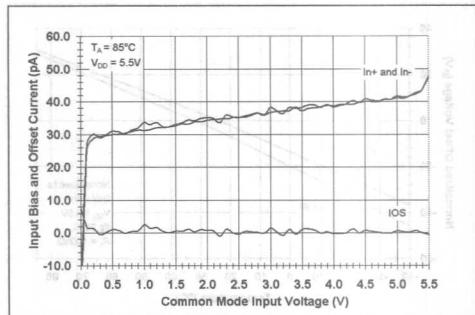


FIGURE 2-16: Input Bias Current, Input Offset Current vs. Common mode Input Voltage

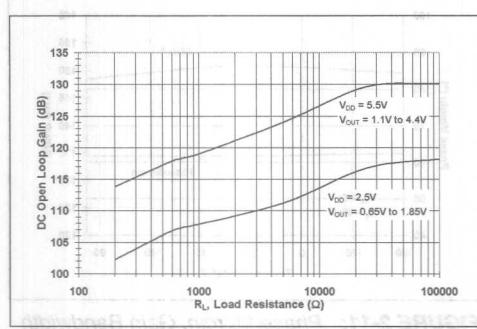


FIGURE 2-14: DC Open Loop Gain vs. Output Load Resistance

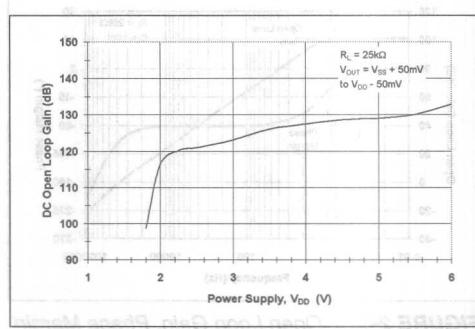


FIGURE 2-17: DC Open Loop Gain vs. Power Supply Voltage

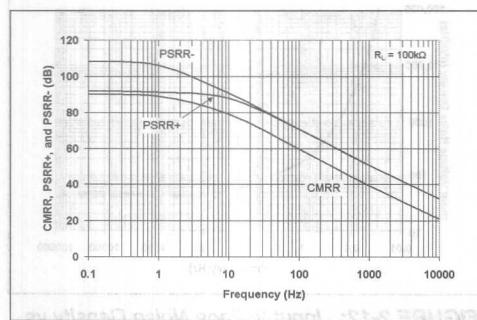


FIGURE 2-15: Common-Mode Rejection Ratio, Power Supply Rejection Ratio vs. Frequency

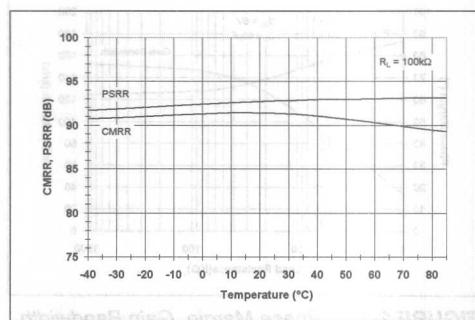


FIGURE 2-18: Common-Mode Rejection Ratio, Power Supply Rejection Ratio vs. Temperature

MCP606/607/608/609

Note: Unless otherwise indicated, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$, $V_{SS} = \text{GND}$

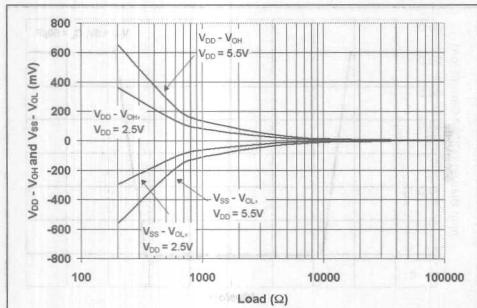


FIGURE 2-19: Low Level and High Level Output Swing vs. Load Resistance

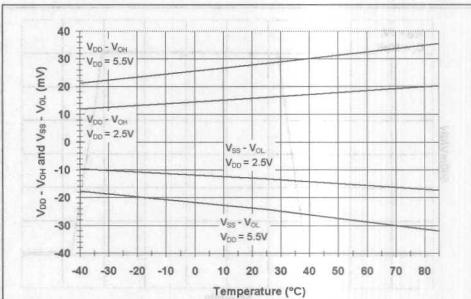


FIGURE 2-22: Low Level and High Level Output Swing vs. Temperature, $R_L = 5\text{k}\Omega$

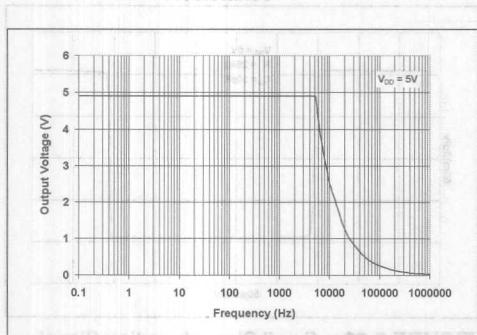


FIGURE 2-20: Maximum Full Scale Output Voltage Swing vs. Frequency

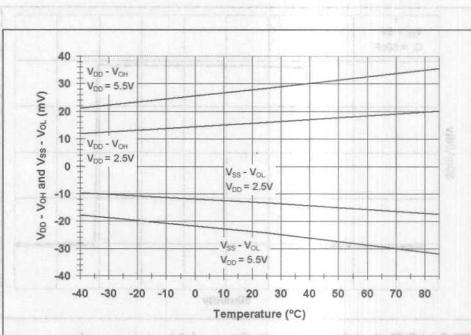


FIGURE 2-23: Low Level and High level Output Swing vs. Temperature, $R_L = 25\text{k}\Omega$

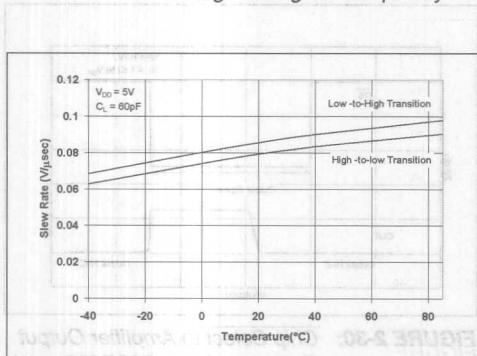


FIGURE 2-21: Slew Rate vs. Temperature

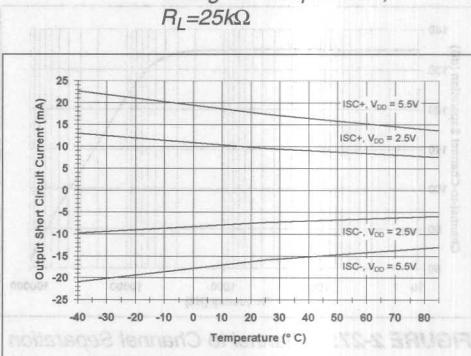


FIGURE 2-24: Output Short Circuit Current vs. Temperature

3

Datasheets

MCP606/607/608/609

Note: Unless otherwise indicated, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$, $V_{SS} = \text{GND}$

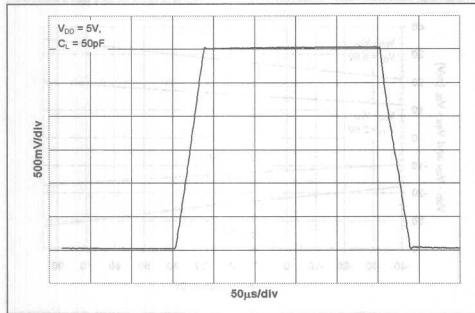


FIGURE 2-25: Large Signal Non-inverting Signal Pulse Response

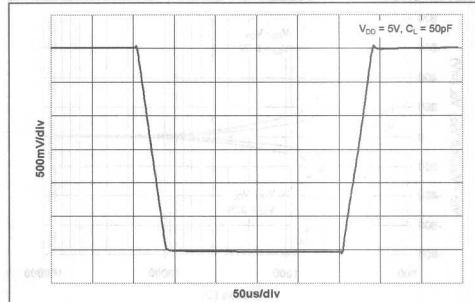


FIGURE 2-28: Large Signal Inverting Signal Pulse Response

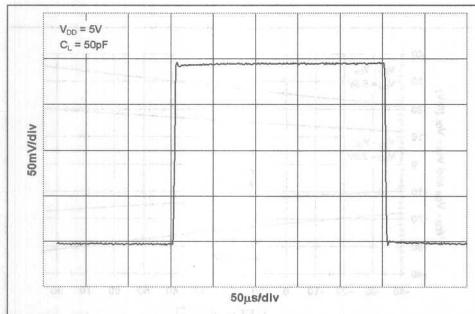


FIGURE 2-26: Small Signal Non-inverting Pulse Response

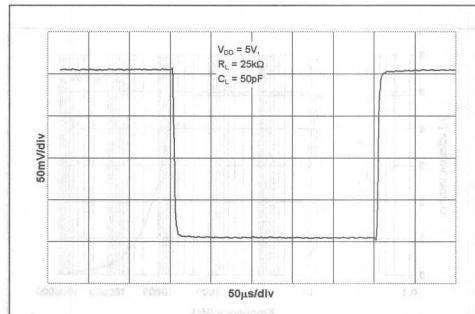


FIGURE 2-29: Small Signal Inverting Signal Pulse Response

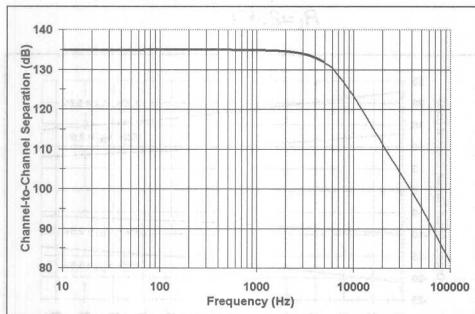


FIGURE 2-27: Channel-to-Channel Separation (MCP607 and MCP609 only)

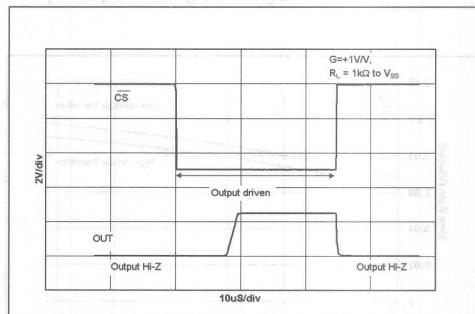


FIGURE 2-30: Chip Select to Amplifier Output Response Time (MCP608 only)

Note: Unless otherwise indicated, $T_A = 25^\circ\text{C}$, $V_{CM} = V_{DD}/2$ and $V_{OUT} \sim V_{DD}/2$, $V_{SS} = \text{GND}$

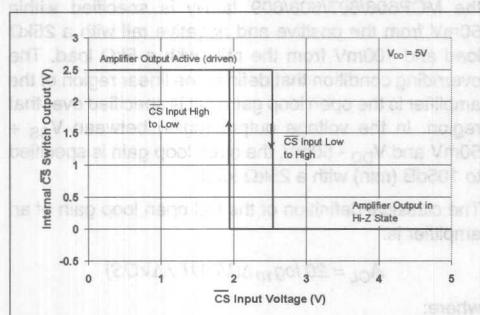


FIGURE 2-31: CS hysteresis (MCP608 only)

Output voltage range of V_{OUT} ($V_{DD} + 30\text{mV}$) to $V_{DD} - 30\text{mV}$ at $V_{SS} = \text{GND}$

Output voltage range of V_{OUT} ($V_{DD} + 30\text{mV}$) to $V_{DD} - 30\text{mV}$ at $V_{SS} = 1.5\text{V}$

Output voltage range of V_{OUT} ($V_{DD} + 30\text{mV}$) to $V_{DD} - 30\text{mV}$ at $V_{SS} = 2.5\text{V}$

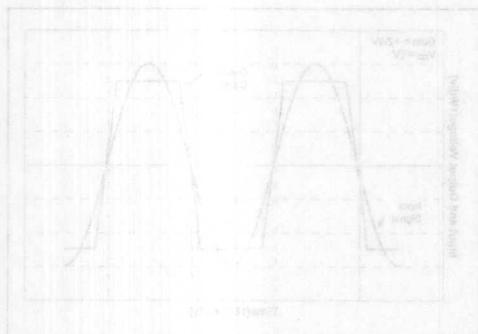


FIGURE 2-32: The MCP608 output voltage vs frequency for $V_{DD} = 2.5\text{V}$ and 5V

The MCP608 has a low noise CMOS input stage. This stage has a high input impedance and a low noise floor. The noise level is approximately $1\text{nV}/\sqrt{\text{Hz}}$ at 100Hz . The noise level is dependent on the supply voltage and the operating temperature. The noise level is also affected by the input signal level. The noise level is approximately $1\text{nV}/\sqrt{\text{Hz}}$ at 100Hz .

The MCP608 has a low noise CMOS input stage. This stage has a high input impedance and a low noise floor. The noise level is approximately $1\text{nV}/\sqrt{\text{Hz}}$ at 100Hz . The noise level is dependent on the supply voltage and the operating temperature. The noise level is also affected by the input signal level. The noise level is approximately $1\text{nV}/\sqrt{\text{Hz}}$ at 100Hz .

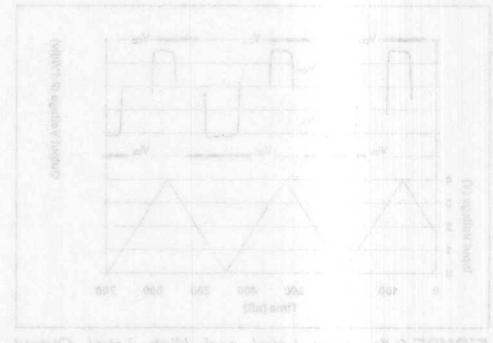


FIGURE 2-33: The MCP608 output voltage vs frequency for $V_{DD} = 1.5\text{V}$, 2.5V , and 5V

MCP606/607/608/609

3.0 APPLICATIONS INFORMATION

The MCP606/607/608/609 family of operational amplifiers are fabricated on Microchip's state-of-the-art CMOS process. They are unity gain stable and suitable for a wide range of general purpose applications. With this family of operational amplifiers, the power supply pin should be by-passed with a $0.1\mu F$ capacitor.

3.1 Rail-to-Rail Output Swing

There are two specifications that describe the output swing capability of the MCP606/607/608/609 family of operational amplifiers. The first specification, Low Level and High Level Output Voltage Swing, defines the absolute maximum swing that can be achieved under specified loaded conditions. For instance, the Low Level Output Voltage Swing of the MCP606/607/608/609 family is specified to be able to swing at least to $15mV$ from the negative rail with a $25k\Omega$ load to $V_{DD}/2$.

This output swing performance is shown in Figure 3-1, where the output of an MCP606 is configured in a gain of $+2V/V$ and overdriven with a 4kHz triangle wave. In this figure, the degradation of the output swing linearity is clearly illustrated. This degradation occurs after the point at which the open loop gain of the amplifier is specified and before the amplifier reaches its maximum and minimum output swing.

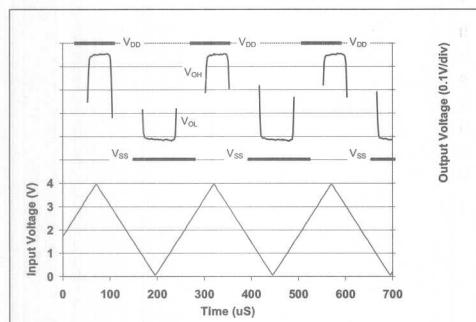


FIGURE 3-1: Low Level and High Level Output Swing

The second specification that describes the output swing capability of these amplifiers is the Linear Region Maximum Output Voltage Swing. This specification defines the maximum output swing that can be achieved while the amplifier is still operating in its linear region.

The Linear Region Maximum Output Voltage Swing of the MCP606/607/608/609 family is specified within $50mV$ from the positive and negative rail with a $25k\Omega$ load and $100mV$ from the rails with a $5k\Omega$ load. The overriding condition that defines the linear region of the amplifier is the open loop gain that is specified over that region. In the voltage output region between $V_{SS} + 50mV$ and $V_{DD} - 50mV$, the open loop gain is specified to $105dB$ (min) with a $25k\Omega$ load.

The classical definition of the DC open loop gain of an amplifier is:

$$A_{OL} = 20 \log_{10} (\Delta V_{OUT} / \Delta V_{OS})$$

where:

A_{OL} is the DC open loop gain of the amplifier,

ΔV_{OUT} is equal to $(V_{DD} - 50mV) - (V_{SS} + 50mV)$ for $R_L = 25k\Omega$, and

ΔV_{OS} is the change in offset voltage with the changing output voltage of the amplifier.

3.2 Input Voltage and Phase Reversal

Since the MCP606/607/608/609 amplifier family is designed with CMOS devices, it does not exhibit phase inversion when the input pins exceed the negative supply voltage. Figure 3-2 shows an input voltage exceeding both supplies with no resulting phase inversion.

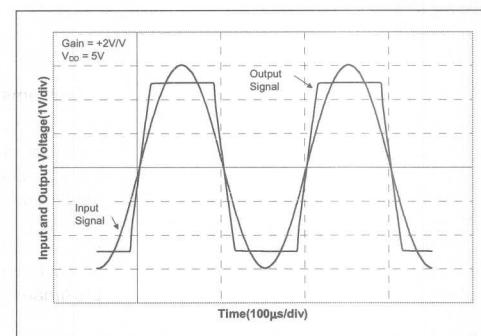


FIGURE 3-2: The MCP606/607/608/609 family of op amps do not have phase reversal issues. For this graph, the amplifier is in a gain of $+2V/V$.

The maximum operating common-mode voltage that can be applied to the inputs is $V_{SS} - 0.3V$ to $V_{DD} + 1.1V$. In contrast, the absolute maximum input voltage is $V_{SS} - 0.3V$ and $V_{DD} + 0.3V$. Voltages on the input that exceed this absolute maximum rating can cause excessive current to flow in or out of the input pins. Current beyond $\pm 2mA$ can cause possible reliability problems. Applications that exceed this rating must be externally limited with an input resistor as shown in Figure 3-3.

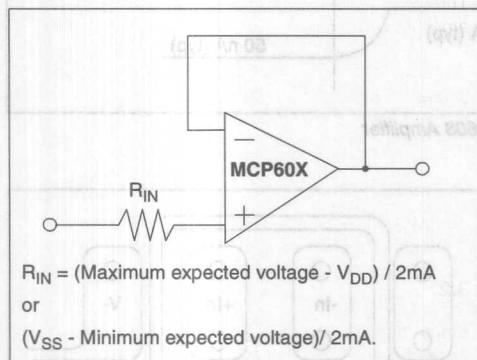


FIGURE 3-3: If the inputs of the amplifier exceed the Absolute Maximum Specifications, an input resistor, R_{IN} , should be used to limit the current flow into that pin.

3.3 Capacitive Load and Stability

Driving capacitive loads can cause stability problems with many of the higher speed amplifiers.

For any closed loop amplifier circuit, a good rule of thumb is to design for a phase margin that is no less than 45° . This is a conservative theoretical value, however, if the phase margin is lower, layout parasitics can degrade the phase margin further causing a truly unstable circuit. A system phase shift of 45° will have an overshoot in its step response of approximately 25%.

A buffer configuration with a capacitive load is the most difficult configuration for an amplifier to maintain stability. The Phase versus Capacitive Load of the MCP60X amplifier is shown in Figure 3-4. In this figure, it can be seen that the amplifier has a phase margin above 40° , while driving capacitance loads up to 220pF.

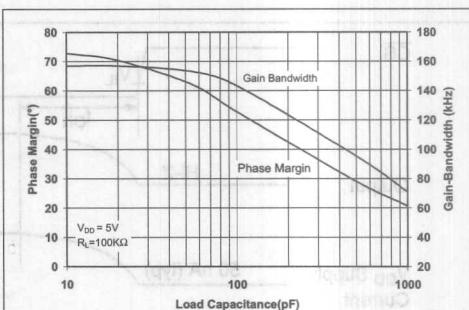


FIGURE 3-4: Gain Bandwidth, Phase Margin vs. Capacitive Load

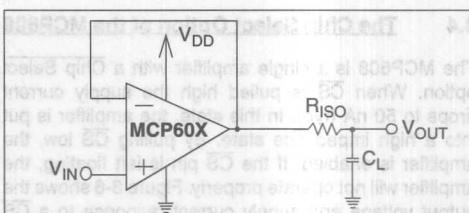


FIGURE 3-5: Amplifier circuits that can be used when driving heavy capacitive loads.

If the amplifier is required to drive larger capacitive loads, the circuit shown in Figure 3-5 can be used. A small series resistor (R_{ISO}) at the output of the amplifier improves the phase margin when driving large capacitive loads. This resistor decouples the capacitive load from the amplifier by introducing a zero in the transfer function.

This zero adjusts the phase margin by approximately:

$$\Delta\theta_m = \tan^{-1}(2\pi GBWP \times R_{ISO} \times C_L)$$

where:

$\Delta\theta_m$ is the improvement in phase margin,

$GBWP$ is the gain bandwidth product of the amplifier,

R_{ISO} is the capacitive decoupling resistor, and

C_L is the load capacitance

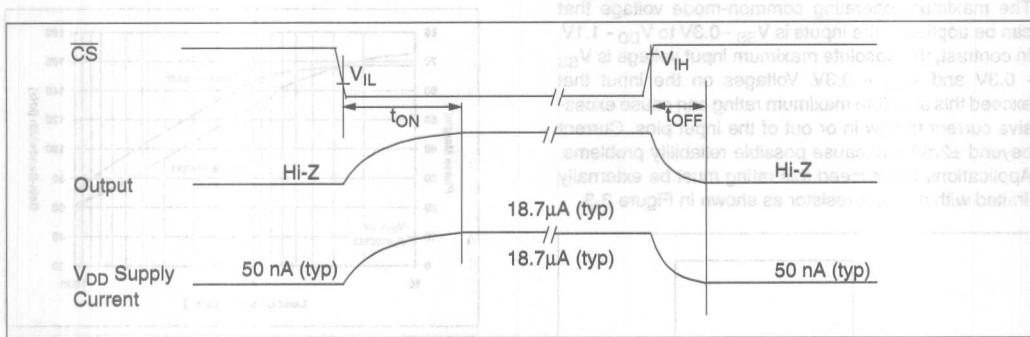


FIGURE 3-6: Timing Diagram for the CS Function of the MCP608 Amplifier

3.4 The Chip Select Option of the MCP608

The MCP608 is a single amplifier with a Chip Select option. When CS is pulled high the supply current drops to 50 nA (typ). In this state, the amplifier is put into a high impedance state. By pulling CS low, the amplifier is enabled. If the CS pin is left floating, the amplifier will not operate properly. Figure 3-6 shows the output voltage and supply current response to a CS pulse.

3.5 Layout Considerations

In applications where low input bias current is critical, PC board surface leakage effects and signal coupling from trace to trace need to be taken into consideration.

3.5.1 SURFACE LEAKAGE

Surface leakage across a PC board is a consequence of differing DC voltages between two traces combined with high humidity, dust or contamination on the board. For instance, the typical resistance from PC board trace to pad is approximately $10^{12}\Omega$ under low humidity conditions. If an adjacent trace is biased to 5V and the input pin of the amplifier is biased at or near zero volts, a 5pA leakage current will appear on the amplifier's input node. This type of PCB leakage is five times the room temperature input bias current (1pA, typ) of the MCP606/607/608/609 family of amplifiers.

The simplest technique that can be used to reduce the effects of PC board leakage is to design a ring around sensitive pins and traces. An example of this type of layout is shown in Figure 3-7.

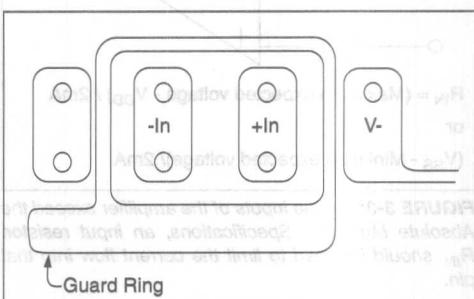


FIGURE 3-7: Example of Guard Ring for the MCP606, the A-amplifier of the MCP607 or the MCP608 in a PC Board Layout

Circuit examples of ring implementations are shown in Figure 3-8. In Figure 3-8A, B and C, the guard ring is biased to the common-mode voltage of the amplifier. This type of guard ring is most effective for applications where the common-mode voltage of the input stage changes, such as buffers, non-inverting gain amplifiers or instrumentation amplifiers.

The strategy shown in Figure 3-8D, biases the common-mode voltage and guard ring to ground. This type of guard ring is typically used in precision photo sensing circuits.

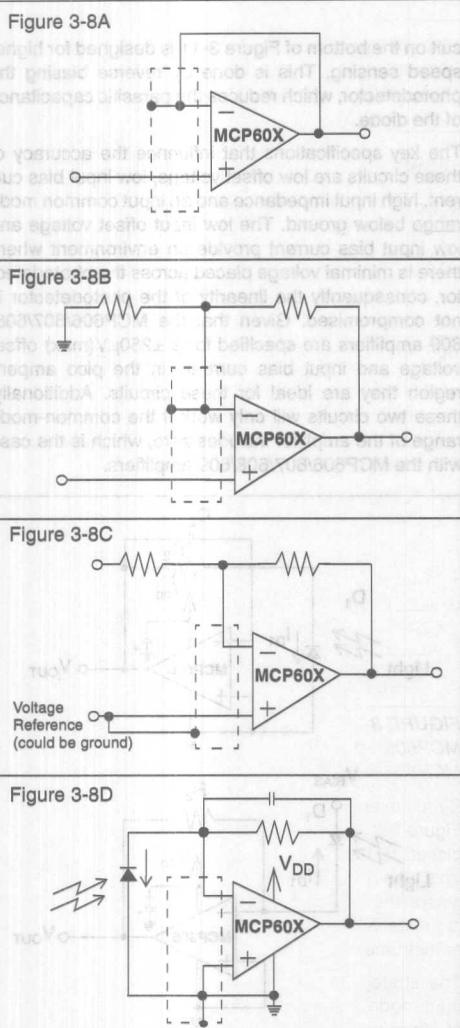


FIGURE 3-8: Examples of how to design PC Board traces to minimize leakage paths to the high impedance input pins of the MCP606/607/608/609 amplifiers.

3.5.2 SIGNAL COUPLING

The input pins of the MCP606/607/608/609 amplifiers have a high impedance providing an opportunity for noise injection, if layout issues are not considered. These high impedance input terminals are sensitive to injected currents. This can occur if the trace from a high impedance input is next to a trace that has fast changing voltages, such as a digital or clock signal. When a high impedance trace is in close proximity to a trace with these types of voltage changes, charge is capacitively coupled into the high impedance trace.

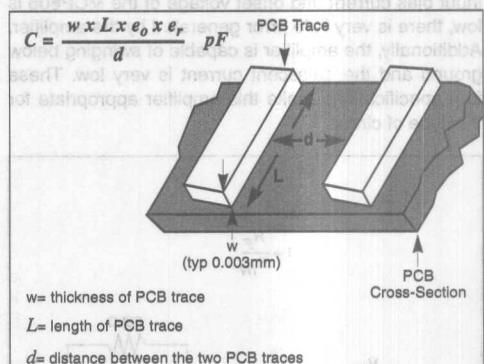


FIGURE 3-9: Capacitors can be built with PCB traces allowing for coupling of signals from one trace to another.

As shown in Figure 3-9, the value of the capacitance between two traces is primarily dependent on the distance (d) between the traces and the distance that the two traces are in parallel (L). From this model, the amount of current generated into the high impedance trace is equal to:

$$I = C \frac{\partial V}{\partial t}$$

where:

I equals the current that appears on the high impedance trace,

C equals the value of capacitance between the two PCB traces,

∂V equals the change in voltage of the trace that is switching, and

∂t equals the amount of time that the voltage change took to get from one level to the next.

MCP606/607/608/609

3.6 Typical Applications

3.6.1 LOW SIDE BATTERY CURRENT SENSING
 The MCP606/607/608/609 amplifiers can be used to sense the output current on the low side of a battery using the circuit in Figure 3-10. In this circuit, the current from the power supply (minus the current required to power the MCP606) flows through a ten ohm resistor from the rest of the circuit in the system. This current is converted to a voltage through the sense resistor and gained by the resistors around the amplifier. Since the input bias current and offset voltage of the MCP606 is low, there is very little error generated by the amplifier. Additionally, the amplifier is capable of swinging below ground and the quiescent current is very low. These four specifications make this amplifier appropriate for this type of circuit.

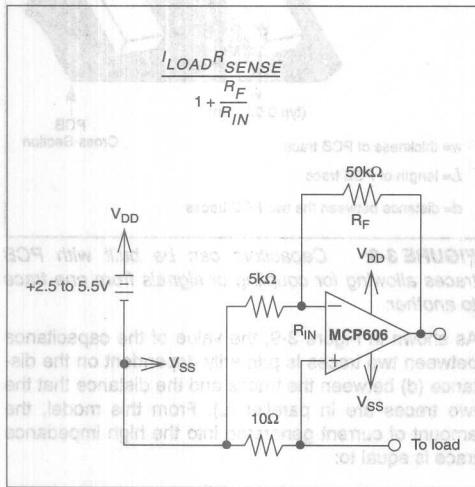


FIGURE 3-10: Low Side Battery Current Sensing

3.6.2 PREAMPLIFIER FOR PHOTO DETECTION CIRCUIT

Any amplifier from this family of operational amplifiers can be used to convert an output current signal from a sensor into a voltage. A sensor that fits this description is a photodetector as shown in Figure 3-11. This type of circuit is implemented with a single resistor and an optional capacitor in the feedback loop of the amplifier. As light impinges on the photo diode, charge is generated, causing a current to flow in the reverse bias direction of the photodetector.

Two circuits are shown in Figure 3-11. The top circuit is designed to provide precision sensing from the photodetector. In this circuit the voltage across the detector is nearly zero and equal to the offset voltage of the amplifier. With this configuration, current that appears across the resistor, R_2 , is primarily a result of the light excitation on the photodiode. The photosensing cir-

cuit on the bottom of Figure 3-11 is designed for higher speed sensing. This is done by reverse biasing the photodetector, which reduces the parasitic capacitance of the diode.

The key specifications that influence the accuracy of these circuits are low offset voltage, low input bias current, high input impedance and an input common mode range below ground. The low input offset voltage and low input bias current provide an environment where there is minimal voltage placed across the photodetector, consequently the linearity of the photodetector is not compromised. Given that the MCP606/607/608/609 amplifiers are specified for a $\pm 250\mu\text{V}(\text{max})$ offset voltage and input bias currents in the pico ampere region they are ideal for these circuits. Additionally, these two circuits will only work if the common-mode range of the amplifier includes zero, which is the case with the MCP606/607/608/609 amplifiers.

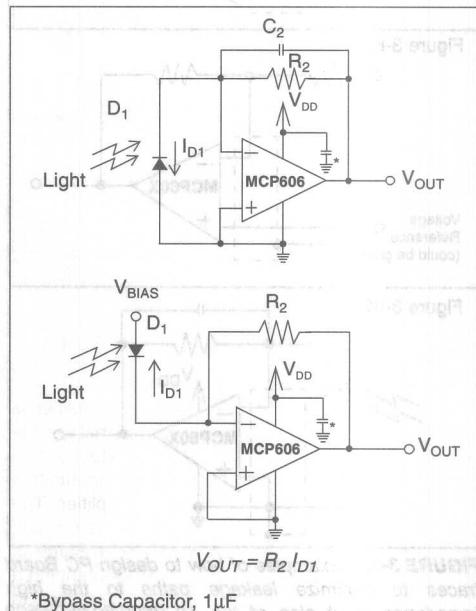


FIGURE 3-11: Pre-Amplifier for Photo Detection Circuit

3.6.3 TWO OP-AMP INSTRUMENTATION AMPLIFIER

The two op-amp instrumentation amplifier shown in Figure 3-12 serves the function of taking the difference of two input voltages, level shifting them and providing a single output. This configuration is best suited for higher gains. (gain > 3 V/V) is shown in Figure 3-12. The key specifications that make the MCP606/607/608/609 family appropriate for this application circuit is low input bias current, low offset voltage and high common-mode rejection. The reference voltage of this circuit is supplied to the first op amp in the signal chain. Typically, this voltage is half of the supply voltage in a single supply environment.

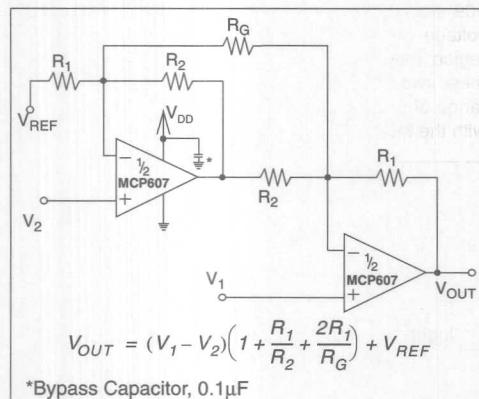


FIGURE 3-12: Two Op-Amp Instrumentation Amplifier

3.6.4 THREE OP-AMP INSTRUMENTATION AMPLIFIER

A classic, three op amp instrumentation amplifier is illustrated in Figure 3-15. The input operational amplifiers provide signal gain. The output operational amplifier converts the signal from two inputs to a single ended output with a difference amplifier. The gain of this circuit is simply adjusted with one resistor, R_G . The reference voltage of the difference stage of this instrumentation amplifier is capable of spanning a wide range. Most typically this node is referenced to half of the supply voltage in a signal supply application.

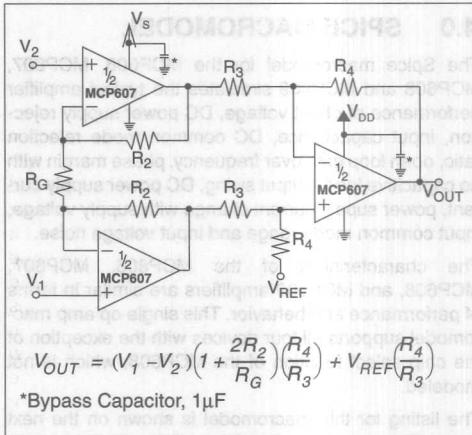


FIGURE 3-13: Three Op-Amp Instrumentation Amplifier

3.6.5 PRECISION GAIN WITH GOOD LOAD ISOLATION

In Figure 3-14, the low input offset voltage of the MCP606 is used to implement a circuit with a high gain. This precision measurement can easily be disrupted by changing the output current drive of the device that is doing the amplification work. Consequently the precision amplifier configuration is followed by a MCP601 amplifier which is capable of driving higher currents. Since the two amplifiers are housed in separate packages, there is minimal change in offset voltage of the MCP606 due to loading effects.

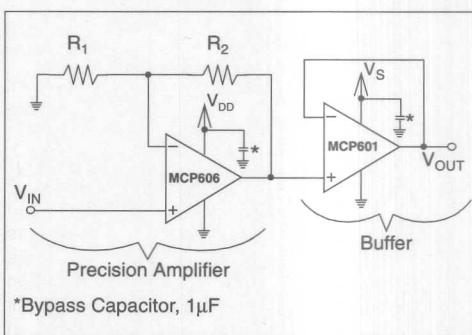


FIGURE 3-14: Precision Gain with Good Load Isolation

MCP606/607/608/609

4.0 SPICE MACROMODEL

The Spice macromodel for the MCP606, MCP607, MCP608 and MCP609 simulates the typical amplifier performance of offset voltage, DC power supply rejection, input capacitance, DC common mode rejection ratio, open loop gain over frequency, phase margin with no capacitive load, output swing, DC power supply current, power supply current change with supply voltage, input common mode range and input voltage noise.

The characteristics of the MCP606, MCP607, MCP608, and MCP609 amplifiers are similar in terms of performance and behavior. This single op amp macro-model supports all four devices with the exception of the chip select function of the MCP608, which is not modeled.

The listing for this macromodel is shown on the next page. The most recent revision of the model can be downloaded from Microchip's web site at www.microchip.com

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the "Company") for its PICmicro® Microcontroller is intended and supplied to you, the Company's customer, for use solely and exclusively on Microchip PICmicro Microcontroller products.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

```
.subckt mcp606 1 2 3 4 5
*          | | | | |
*          | | | | Output
*          | | | Negative supply
*          | | Positive Supply
*          | Inverting input
*          Non-inverting input
*
* Macromodel for MCP606 (single), MCP607 (dual), MCP608 (single w/CS), and MCP609 (quad)
*
* The characteristics of the MCP606, MCP607, MCP608, and MCP609 have the same fundamental
* performance and behavior. Consequently, this single op amp macromodel supports all four
* devices. However, the chip select function of the MCP608 is not modeled.
*
* Revision History:
*   REV A : 6-30-99 created BCB
*
* This macromodel models typical amplifier offset voltage, DC power supply rejection, input
* capacitance, open loop gain over frequency, phase margin with 60pF load, output swing,
* power supply current, input voltage noise, slew rate.
*
* NOTICE: THE INFORMATION PROVIDED HEREIN IS BELIEVED TO BE RELIABLE,
* HOWEVER, MICROCHIP ASSUMES NO RESPONSIBILITY FOR INACCURACIES OR
* OMISSIONS. MICROCHIP ASSUMES NO RESPONSIBILITY FOR THE USE OF THIS
* INFORMATION, AND ALL USE OF SUCH INFORMATION SHALL BE ENTIRELY AT
* THE USER'S OWN RISK. NO INTELLECTUAL PROPERTY RIGHTS OR LICENSES
* TO ANY OF THE TECHNOLOGY DESCRIBED HEREIN ARE IMPLIED OR GRANTED TO
* ANY THIRD PARTY. MICROCHIP RESERVES THE RIGHT TO CHANGE THIS MODEL
* AT ANY TIME WITHOUT NOTICE.
*
*
*Input Stage, pole at 300kHz
M1    9      64    7      3      Ptype
M2    8      2      7      3      Ptype
CDIFF  1      2      3E-12
CCM1   1      4      6E-12
CCM2   2      4      6E-12
IDD    3      7      13.33e-6
RA     8      6      1.839e3
RB     9      6      1.839e3
CA     8      9      125e-12
ICOMP  3      4      -194.63e-6
*
*Input Stage Common-Mode Clamping
VCMM   4      6      0.35
ECM    55     4      3 64 1
RCM    57     56    1E3
DCMP   56     55    DY
VCMP   57     4      1.2
```

```

RST 58 59 1E3
DST 59 55 DX
VST 58 4 1.6
GCMP2 23 4 POLY(2) 57 56 58 59 0 0 0;0 -0.5E-3 0.5E-3

*Input errors (vos, en, psr)

ERR 64 1 poly(2) (67,4) (3, 4) -229.9e-6 1 23e-6

*Second Stage, pole at 0.183Hz
GS 23 4 8 9 543.78e-6
R1 23 4 8.2144e9
C2 23 4 110e-12

VSOM 3 24 4.784
VSOP 25 4 -3.98
DSOM 23 24 DY
DSOP 25 23 DY

*HCM 23 3 VCOMP

FS 3 4 POLY(11) V03 V05 V04 V06 V01 V02 V09 V010 VMID1 VSOP VSOM
+ 200E-6 -1 -1 -1 1 -1 -1 1 1 -1 -1 -1

*mid-supply reference
RMID1 3 35 61.62E3
VMID1 35 34 0
RMID2 4 34 61.62E3
ELEVEL 34 4 23 4 -1

*DOutput stage
DO3 34 43 DY
DO4 44 34 DY
DO5 3 45 DY
DO6 3 46 DY
DO7 4 45 DY
DO8 4 46 DY
VO3 43 5 0.1
VO4 5 44 0.03
G05 3 47 3 34 10E-3
VO5 47 5 0
G06 4 48 34 4 10E-3
VO6 48 5 0
G01 49 4 5 34 10E-3
VO1 49 45 0
G02 50 4 34 5 10E-3
VO2 50 46 0
RO9 3 51 100
VO9 51 5 0
RO10 52 4 100
VO10 52 5 0

* input voltage noise
VN1 65 4 0.6
DN1 65 67 DX
RN1 67 4 10E3

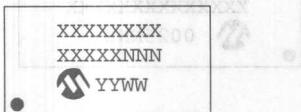
.model Ptype PMOS L=2 W=105
.model DY D (IS=1e-15 BV =50)
.model DX D (IS=1e-18 AF=0.6 KF=10e-17)
.ENDS

```

5.0 PACKAGING INFORMATION

5.1 Package Marking Information

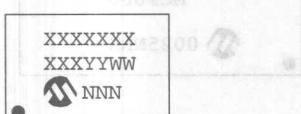
8-Lead PDIP (300 mil)



Example

MCP606
XXXXX
YYWW
NNN

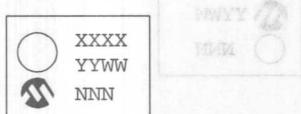
8-Lead SOIC (208 mil)



Example

MCP606
XXXXX
YYWW
NNN

8-Lead TSSOP



Example

MCP606
XXXXX
YYWW
NNN

Legend: XX...X Customer specific information*

YY Year code (last 2 digits of calendar year)

WW Week code (week of January 1 is week '01)

NNN Alphanumeric traceability code

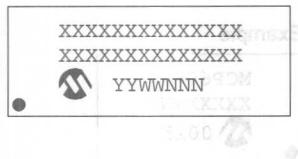
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

MCP606/607/608/609

Package Marking Information (Continued)

14-Lead PDIP (300 mil)



14-Lead SOIC (208 mil)

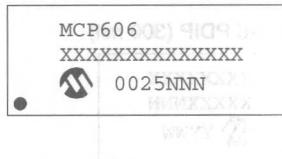


14-Lead TSSOP



PACKAGE MARKING INFORMATION

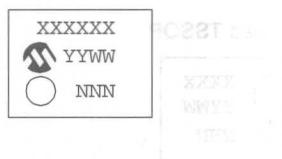
Example



Example



Example



Legend: XX...X Customer specific information*

YY Year code (last 2 digits of calendar year)

WW Week code (week of January 1 is week '01')

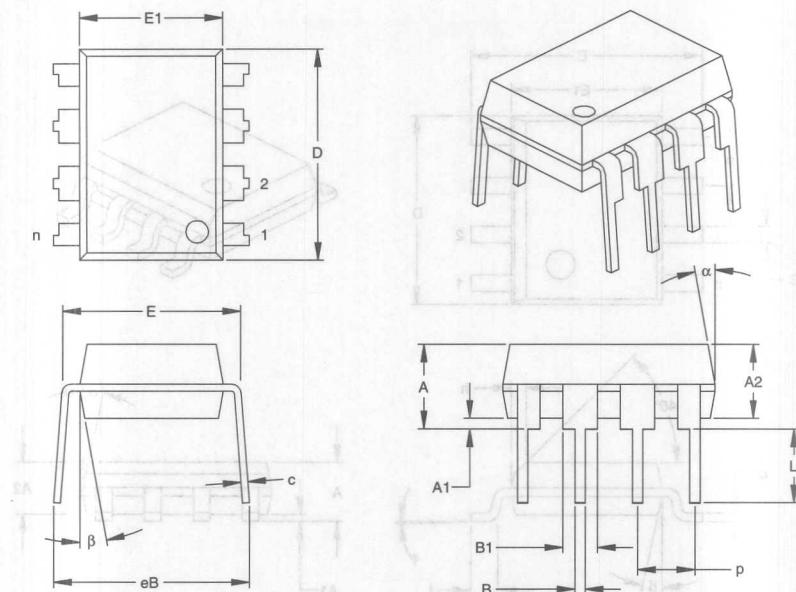
NNN Alphanumeric traceability code (last 3 digits of part number)

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

- * Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

MCP606/607/608/609

8-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Dimension Limits	Units			INCHES*			MILLIMETERS		
	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8					8	
Pitch	p		.100					2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32		
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68		
Base to Seating Plane	A1	.015			0.38				
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26		
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60		
Overall Length	D	.360	.373	.385	9.14	9.46	9.78		
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43		
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38		
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78		
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56		
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40	10.92		
Mold Draft Angle Top	α	5	10	15	5	10	15		
Mold Draft Angle Bottom	β	5	10	15	5	10	15		

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

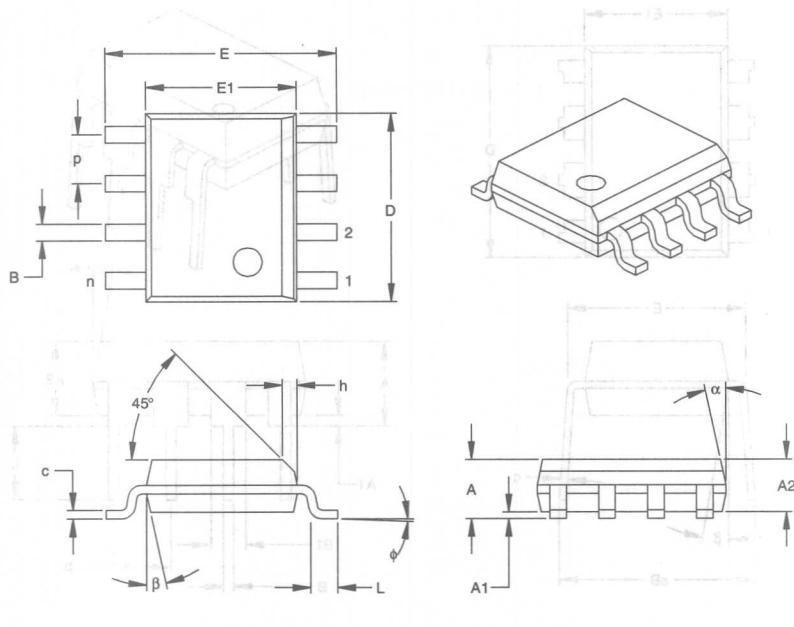
JEDEC Equivalent: MS-001

Drawing No. C04-018

3
Datasheets

MCP606/607/608/609

8-Lead Plastic Small Outline (SN) – Narrow, 150 mil (SOIC) Rev 006 – (3) coil-in lead – (4) lead-0



Dimension	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n	.08	.08	.09	.20	.20	.25
Pitch	p	.050	.050	.050	.127	.127	.127
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	0.10	0.18	0.25
Overall Width	E	.228	.237	.244	5.79	6.02	6.20
Molded Package Width	E1	.146	.154	.157	3.71	3.91	3.99
Overall Length	D	.189	.193	.197	4.80	4.90	5.00
Chamfer Distance	h	.010	.015	.020	0.25	0.38	0.51
Foot Length	L	.019	.025	.030	0.48	0.62	0.76
Foot Angle	φ	0°	4°	8°	0°	4°	8°
Lead Thickness	c	.008	.009	.010	0.20	0.23	0.25
Lead Width	B	.013	.017	.020	0.33	0.42	0.51
Mold Draft Angle Top	α	0°	12°	15°	0°	12°	15°
Mold Draft Angle Bottom	β	0°	12°	15°	0°	12°	15°

* Controlling Parameter

§ Significant Characteristic

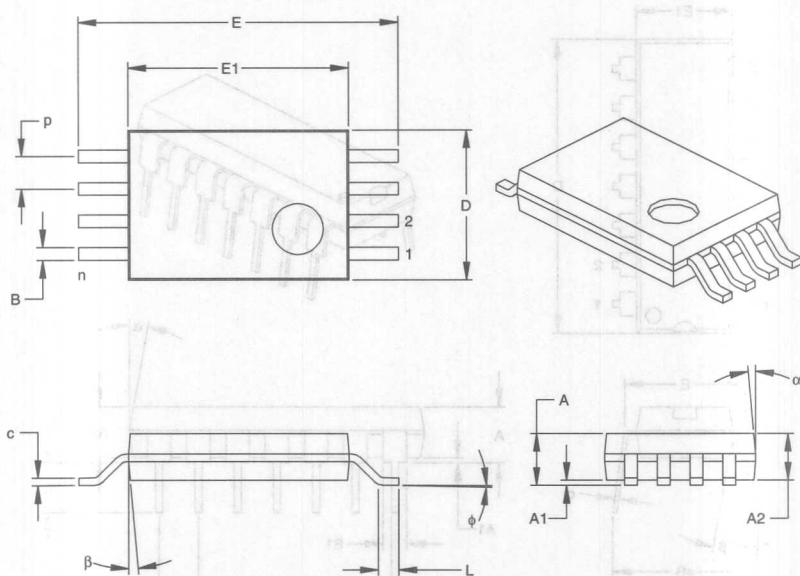
Dimensions D and E do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-012

Drawing No. C04-057

MCP606/607/608/609

8-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP) 2 – (S) anti-infrared version lead-free



Dimension	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	P		.026			0.65	
Overall Height	A			.043			1.10
Molded Package Thickness	A2	.033	.035	.037	0.85	0.90	0.95
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Overall Width	E	.246	.251	.256	6.25	6.38	6.50
Molded Package Width	E1	.169	.173	.177	4.30	4.40	4.50
Molded Package Length	D	.114	.118	.122	2.90	3.00	3.10
Foot Length	L	.020	.024	.028	0.50	0.60	0.70
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.007	.010	.012	0.19	0.25	0.30
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .005" (0.127mm) per side.

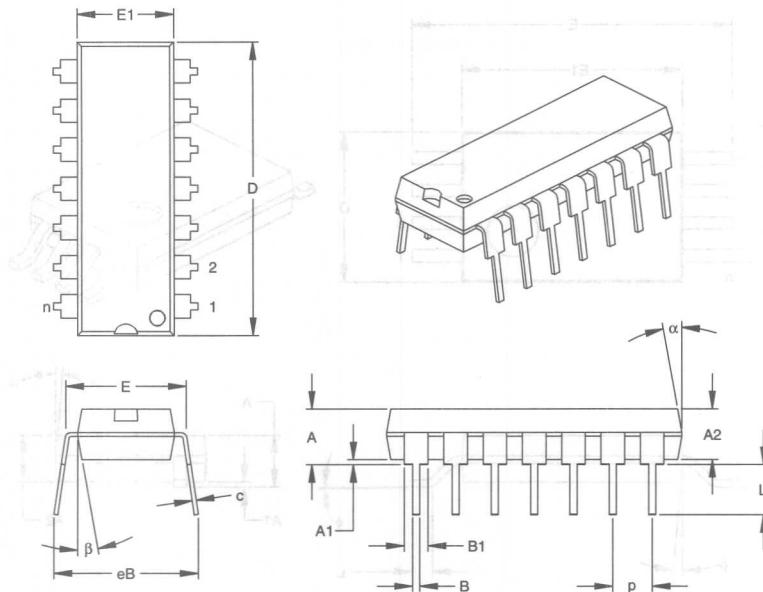
JEDEC Equivalent: MO-153

Drawing No. C04-086

3
Datasheets

MCP606/607/608/609

14-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Dimension Limits	INCHES*			MILLIMETERS		
	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		.14			.14
Pitch	p		.100			2.54
Top to Seating Plane	A	.140	.155	.170	3.56	3.94
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30
Base to Seating Plane	A1	.015			0.38	
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94
Molded Package Width	E1	.240	.250	.260	6.10	6.35
Overall Length	D	.740	.750	.760	18.80	19.05
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30
Lead Thickness	c	.008	.012	.015	0.20	0.29
Upper Lead Width	B1	.045	.058	.070	1.14	1.46
Lower Lead Width	B	.014	.018	.022	0.36	0.46
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40
Mold Draft Angle Top	alpha	5	10	15	5	10
Mold Draft Angle Bottom	beta	5	10	15	5	10

* Controlling Parameter

§ Significant Characteristic

Notes:

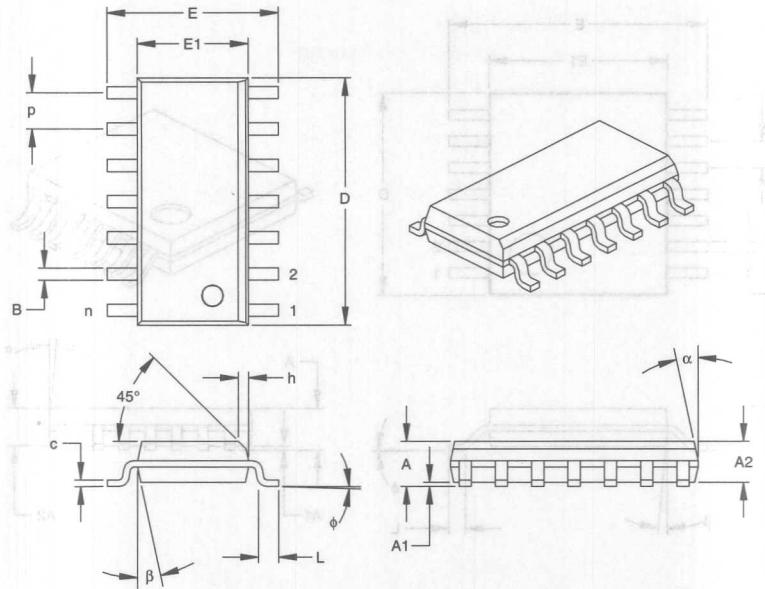
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-005

MCP606/607/608/609

14-Lead Plastic Small Outline (SL) – Narrow, 150 mil (SOIC)



Dimension Limits	INCHES*			MILLIMETERS		
	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		.14			.14
Pitch	p	.050				.127
Overall Height	A	.053	.061	.069	1.35	1.55
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42
Standoff §	A1	.004	.007	.010	0.10	0.18
Overall Width	E	.228	.236	.244	5.79	5.99
Molded Package Width	E1	.150	.154	.157	3.81	3.90
Overall Length	D	.337	.342	.347	8.56	8.69
Chamfer Distance	h	.010	.015	.020	0.25	0.38
Foot Length	L	.016	.033	.050	0.41	0.84
Foot Angle	φ	0	4	8	0	4
Lead Thickness	c	.008	.009	.010	0.20	0.23
Lead Width	B	.014	.017	.020	0.36	0.42
Mold Draft Angle Top	α	0	12	15	0	12
Mold Draft Angle Bottom	β	0	12	15	0	12

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-012

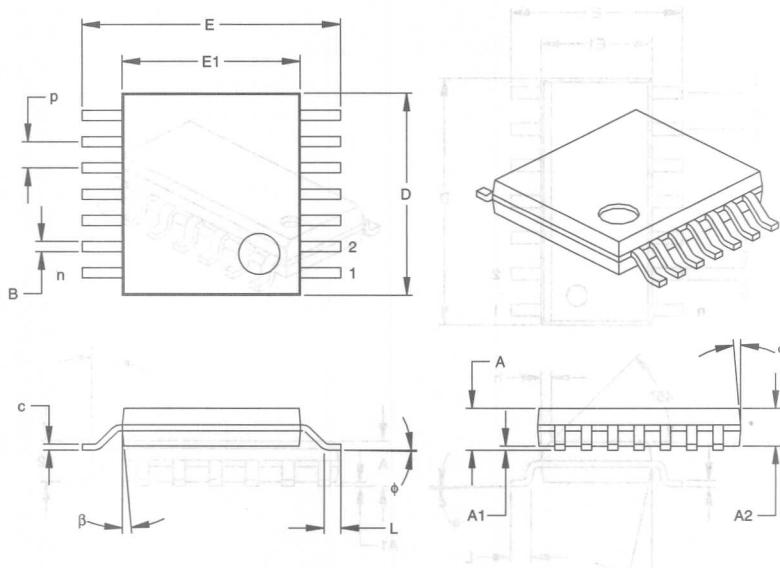
Drawing No. C04-065

3

Datasheets

MCP606/607/608/609

14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP) JEDEC MO-153-1000-001



Dimension	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p	.026				0.65	
Overall Height	A			.043			1.10
Molded Package Thickness	A2	.033	.035	.037	0.85	0.90	0.95
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Overall Width	E	.246	.251	.256	6.25	6.38	6.50
Molded Package Width	E1	.169	.173	.177	4.30	4.40	4.50
Molded Package Length	D	.193	.197	.201	4.90	5.00	5.10
Foot Length	L	.020	.024	.028	0.50	0.60	0.70
Foot Angle	ϕ	0	4	8	0	4	8
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B1	.007	.010	.012	0.19	0.25	0.30
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .005" (0.127mm) per side.

JEDEC Equivalent: MO-153

Drawing No. C04-087

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

<ftp://ftp.microchip.com>

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-786-7302 for the rest of the world.

990902

3

Datasheets

Trademarks: The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER, PRO MATE and MPLAB are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. FlexROM and fuzzyLAB are trademarks and SQTP is a service mark of Microchip in the U.S.A.

All other trademarks mentioned herein are the property of their respective companies.

READER RESPONSE

TRICKS OF THE TRADE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager 480-786-7578

RE: Reader Response 480-786-7578

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) - _____

Total Pages Sent

Application (optional):

Would you like a reply? Y N

Device: MCP606/607/608/609

Literature Number: DS11177B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?

MCP606/607/608/609

MCP606/607/608/609 PRODUCT IDENTIFICATION SYSTEM

:23TOM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

Part Number	X	/X	Package:	P = Plastic DIP (300 mil Body), 8-lead and 14-lead SN = Plastic SOIC (150 mil Body), 8-lead SL = Plastic SOIC (150 mil Body), 14-lead ST = Plastic TSSOP, 8-lead and 14-lead
			Temperature Range:	I = -40°C to +85°C
			Device:	MCP606 = Single Operational Amplifier MCP606T = Single Operational Amplifier (Tape and Reel-SOIC/TSSOP) MCP607 = Dual Operational Amplifier MCP607T = Dual Operational Amplifier (Tape and Reel-SOIC/TSSOP) MCP608 = Single Operational Amplifier w/ \overline{CS} Function MCP608T = Single Operational Amplifier w/ \overline{CS} Function (Tape and Reel-SOIC/TSSOP) MCP609 = Quad Operational Amplifier MCP609T = Quad Operational Amplifier (Tape and Reel-SOIC/TSSOP)

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

3

Datasheets

MCP606/607/608/609

NOTES:

MCP606/607/608/609 PRODUCT IDENTIFICATION SYSTEM

In order to determine the part number, refer to the following letter of the family or the lead frame code.

Part Number	Base code	Temperature Range
MCP606		-40°C to +85°C
MCP606T	T = Single Chip Version	
MCP606A	A = Single Die Version (Type A or B = 300L120)	
MCP606L	L = Dual Die Version (Type L = 300L230)	
MCP606R	R = Dual Die Version (Type R = 300L230)	
MCP606S	S = Single Chip Version (Type S = 300L120)	
MCP606T	T = Single Die Version (Type T = 300L120)	
MCP606A	A = Single Die Version (Type A = 300L120)	
MCP606L	L = Dual Die Version (Type L = 300L230)	
MCP606R	R = Dual Die Version (Type R = 300L230)	

Part Number	Options
MCP606	No options
MCP606A	Die Options
MCP606L	Die Options
MCP606R	Die Options
MCP606S	Die Options
MCP606T	Die Options

Options available for MCP606/607/608/609

- 1. Add suffix "P" to part number to select option
- 2. The MCP606/607/608/609 Options Catalogue is available from Microchip U.S. FAX: (480) 248-7373
- 3. The MCP606/607/608/609 Options Catalogue Site (www.microchip.com)

Please check with your distributor to confirm availability of options (U.S. distributor only).

Microchip Technology Inc. reserves the right to make changes without notice to its products.

Microchip is a registered trademark of Microchip Technology Inc.



MICROCHIP

MCP3001

2.7V 10-Bit A/D Converter with SPI™ Serial Interface

FEATURES

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200ksps sampling rate at 5V
- 75ksps sampling rate at 2.7V
- Low power CMOS technology
 - 5nA typical standby current, 2 μ A max
 - 500 μ A max active current at 5V
- Industrial temp range: -40°C to +85°C
- 8-pin PDIP, SOIC and TSSOP packages

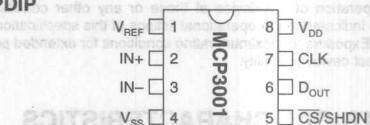
APPLICATIONS

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

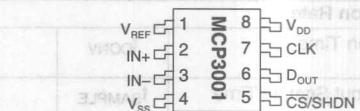
DESCRIPTION

The Microchip Technology Inc. MCP3001 is a successive approximation 10-bit A/D converter (ADC) with on-board sample and hold circuitry. The device provides a single pseudo-differential input. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are both specified at ± 1 LSB max. Communication with the device is done using a simple serial interface compatible with the SPI protocol. The device is capable of sample rates of up to 200ksps at a clock rate of 2.8MHz. The MCP3001 operates over a broad voltage range (2.7V - 5.5V). Low current design permits operation with a typical standby current of only 5nA and a typical active current of 400 μ A. The device is offered in 8 pin PDIP, TSSOP and 150mil SOIC packages.

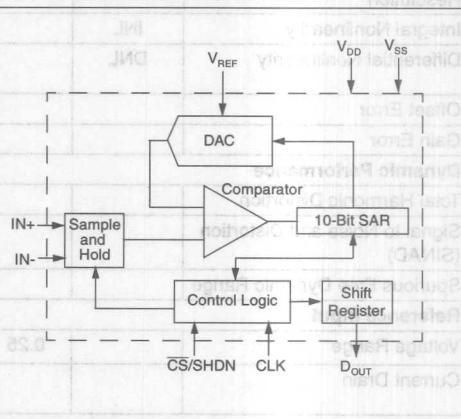
PACKAGE TYPES



SOIC, TSSOP



FUNCTIONAL BLOCK DIAGRAM



SPI is a trademark of Motorola Inc.

MCP3001

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to $V_{DD} + 0.6V$
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	> 4kV

*Notice: Stresses above those listed under "Maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
V_{DD}	+2.7V to 5.5V Power Supply
V_{SS}	Ground
IN+	Positive Analog Input
IN-	Negative Analog Input
CLK	Serial Clock
D_{OUT}	Serial Data Out
CS/SHDN	Chip select/Shutdown Input
V_{REF}	Reference Voltage Input

ELECTRICAL CHARACTERISTICS

All parameters apply at $V_{DD} = 5V$, $V_{SS} = 0V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 200\text{ksps}$ and $f_{CLK} = 14 \cdot f_{SAMPLE}$ unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_{AMB} = 25^{\circ}C$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Conversion Rate						
Conversion Time	t_{CONV}			10	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}			200 75	ksps ksps	$V_{DD} = V_{REF} = 5V$ $V_{DD} = V_{REF} = 2.7V$
DC Accuracy						
Resolution			10		bits	
Integral Nonlinearity	INL		± 0.5	± 1	LSB	
Differential Nonlinearity	DNL		± 0.25	± 1	LSB	No missing codes over temperature
Offset Error				± 1.5	LSB	
Gain Error				± 1	LSB	
Dynamic Performance						
Total Harmonic Distortion			-76		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Signal to Noise and Distortion (SINAD)			61		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Spurious Free Dynamic Range			80		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Reference Input						
Voltage Range		0.25		V_{DD}	V	Note 2
Current Drain			90 0.001	150 3	μA	$\overline{CS} = V_{DD} = 5V$
Analog Inputs						
Input Voltage Range (IN+)		IN-		$V_{REF} + IN -$	V	
Input Voltage Range (IN-)		$V_{SS} - 100$		$V_{SS} + 100$	mV	
Leakage Current			0.001	± 1	μA	
Switch Resistance	R_{SS}		1K		Ω	See Figure 4-1
Sample Capacitor	C_{SAMPLE}		20		pF	See Figure 4-1

ELECTRICAL CHARACTERISTICS (CONTINUED)

All parameters apply at $V_{DD} = 5V$, $V_{SS} = 0V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 200\text{ksps}$ and $f_{CLK} = 14*f_{SAMPLE}$ unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_{AMB} = 25^{\circ}C$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Digital Input/Output						
Data Coding Format						
High Level Input Voltage	V_{IH}	0.7 V_{DD}		—	V	
Low Level Input Voltage	V_{IL}	—		0.3 V_{DD}	V	
High Level Output Voltage	V_{OH}	4.1		—	V	$I_{OH} = -1\text{mA}$, $V_{DD} = 4.5\text{V}$
Low Level Output Voltage	V_{OL}	—		0.4	V	$I_{OL} = 1\text{mA}$, $V_{DD} = 4.5\text{V}$
Input Leakage Current	I_U	-10		10	μA	$V_{IN} = V_{SS}$ or V_{DD}
Output Leakage Current	I_{LO}	-10		10	μA	$V_{OUT} = V_{SS}$ or V_{DD}
Pin Capacitance (all inputs/outputs)	C_{IN}, C_{OUT}	—		10	pF	$V_{DD} = 5.0\text{V}$ (Note 1) $T_{AMB} = 25^{\circ}\text{C}$, $f = 1\text{ MHz}$
Timing Parameters						
Clock Frequency	f_{CLK}			2.8 1.05	MHz MHz	$V_{DD} = 5\text{V}$ (Note 3) $V_{DD} = 2.7\text{V}$ (Note 3)
Clock High Time	t_{H1}	160			ns	
Clock Low Time	t_{L1}	160			ns	
CS Fall To First Rising CLK Edge	t_{SUCS}	100			ns	
CLK Fall To Output Data Valid	t_{DO}			125 200	ns ns	$V_{DD} = 5\text{V}$, See Figure 1-2 $V_{DD} = 2.7$, See Figure 1-2
CLK Fall To Output Enable	t_{EN}			125 200	ns ns	$V_{DD} = 5\text{V}$, See Figure 1-2 $V_{DD} = 2.7$, See Figure 1-2
CS Rise To Output Disable	t_{DIS}			100	ns	See test circuits, Figure 1-2 (Note 1)
CS Disable Time	t_{CSH}	350			ns	
D_{OUT} Rise Time	t_R			100	ns	See test circuits, Figure 1-2 (Note 1)
D_{OUT} Fall Time	t_F			100	ns	See test circuits, Figure 1-2 (Note 1)
Power Requirements						
Operating Voltage	V_{DD}	2.7		5.5	V	
Operating Current	I_{DD}			400 210	μA μA	$V_{DD} = 5.0\text{V}$, D_{OUT} unloaded $V_{DD} = 2.7\text{V}$, D_{OUT} unloaded
Standby Current	I_{DDS}			0.005	2	μA
						$CS = V_{DD} = 5.0\text{V}$

Note 1: This parameter is guaranteed by characterization and not 100% tested.

2: See graph that relates linearity performance to V_{REF} level.

3: Because the sample cap will eventually lose charge, clock rates below 10kHz can affect linearity performance, especially at elevated temperatures.

MCP3001

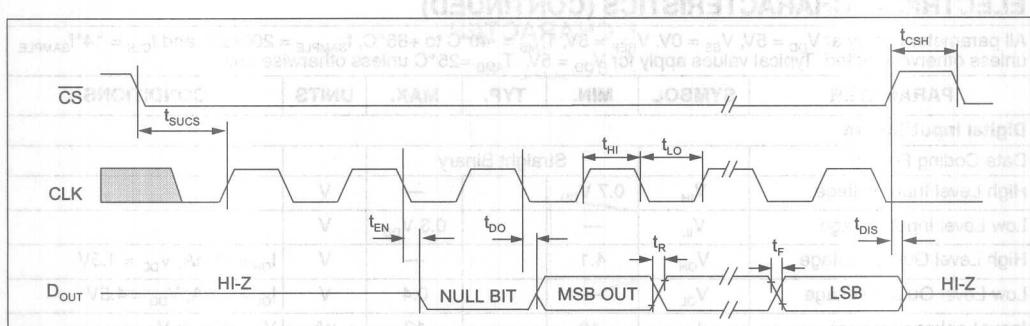


FIGURE 1-1: Serial Timing.

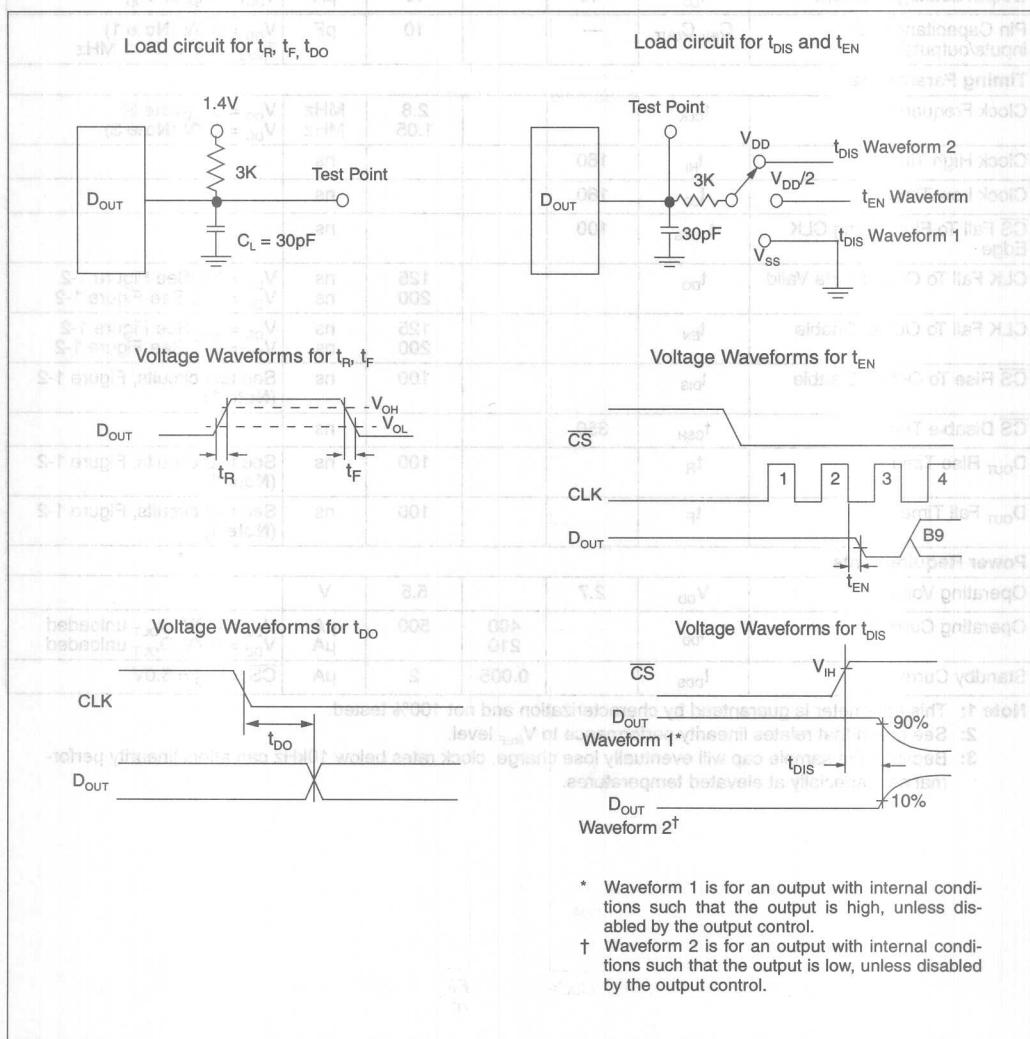


FIGURE 1-2: Test Circuits.

2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 14\text{*Sample Rate}$, $T_A = 25^\circ\text{C}$

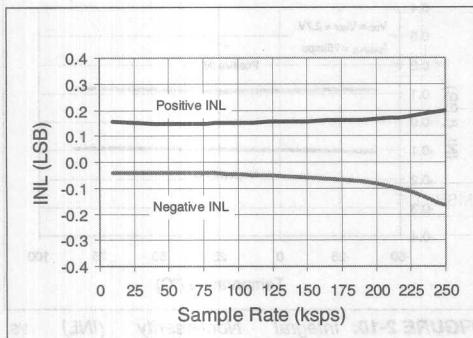


FIGURE 2-1: Integral Nonlinearity (INL) vs. Sample Rate.

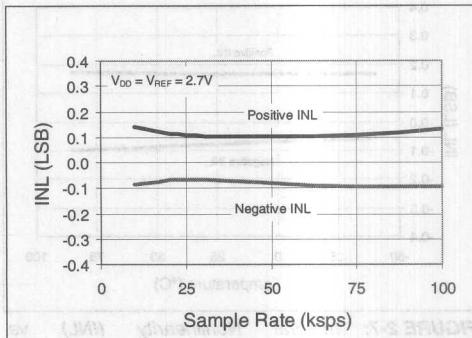


FIGURE 2-4: Integral Nonlinearity (INL) vs. Sample Rate ($V_{DD} = 2.7V$).

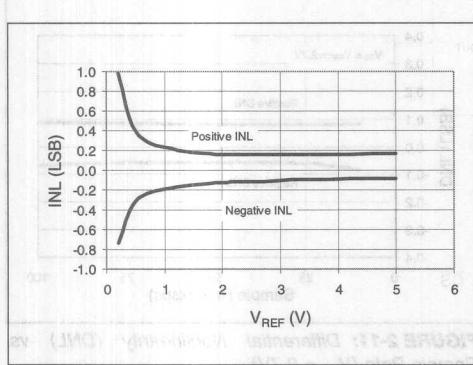


FIGURE 2-2: Integral Nonlinearity (INL) vs. V_{REF} .

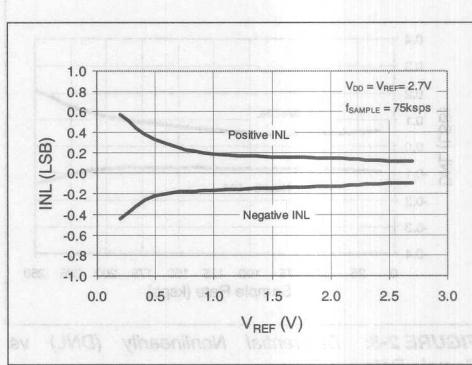


FIGURE 2-5: Integral Nonlinearity (INL) vs. V_{REF} . ($V_{DD} = 2.7V$)

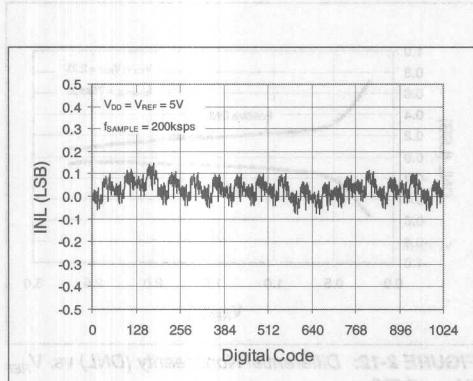


FIGURE 2-3: Integral Nonlinearity (INL) vs. Code (Representative Part).

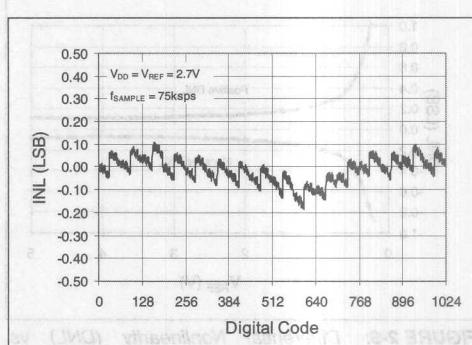


FIGURE 2-6: Integral Nonlinearity (INL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

MCP3001

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 14 \times \text{Sample Rate}$, $T_A = 25^\circ\text{C}$

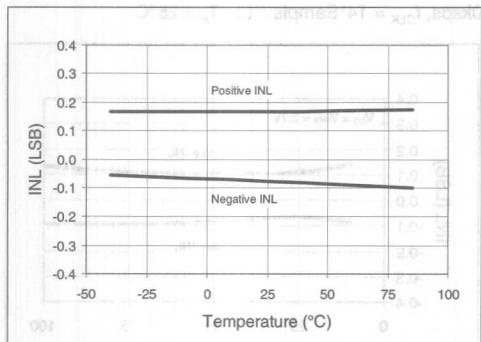


FIGURE 2-7: Integral Nonlinearity (INL) vs. Temperature.

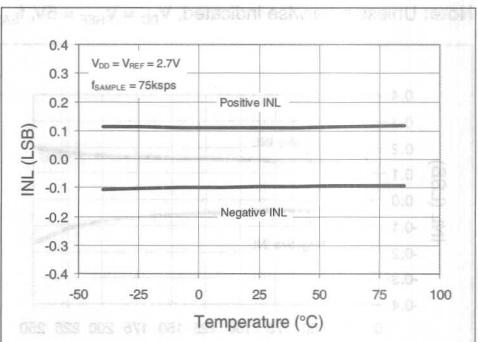


FIGURE 2-10: Integral Nonlinearity (INL) vs. Temperature ($V_{DD} = 2.7V$).

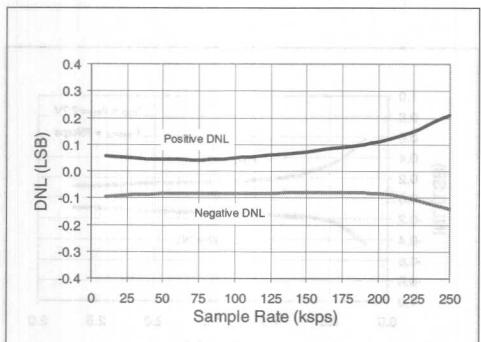


FIGURE 2-8: Differential Nonlinearity (DNL) vs. Sample Rate.

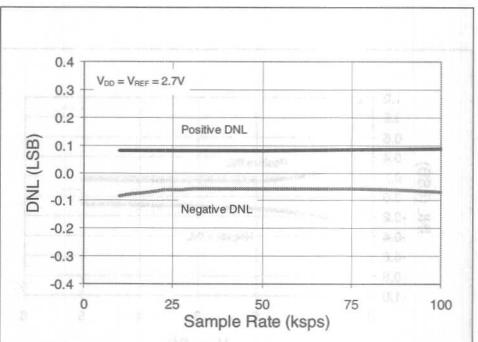


FIGURE 2-11: Differential Nonlinearity (DNL) vs. Sample Rate ($V_{DD} = 2.7V$).

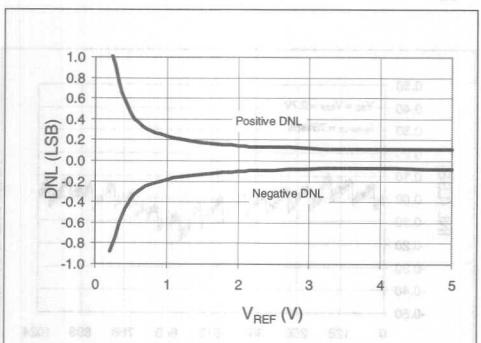


FIGURE 2-9: Differential Nonlinearity (DNL) vs. V_{REF} .

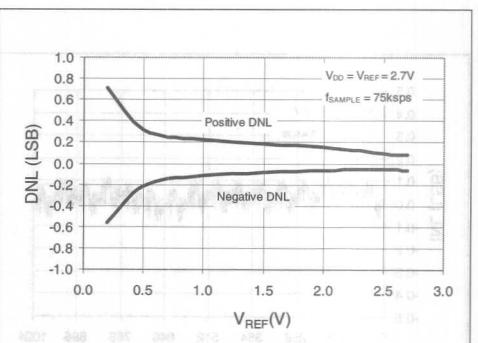


FIGURE 2-12: Differential Nonlinearity (DNL) vs. V_{REF} ($V_{DD} = 2.7V$).

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 14 \times \text{Sample Rate}$, $T_A = 25^\circ\text{C}$

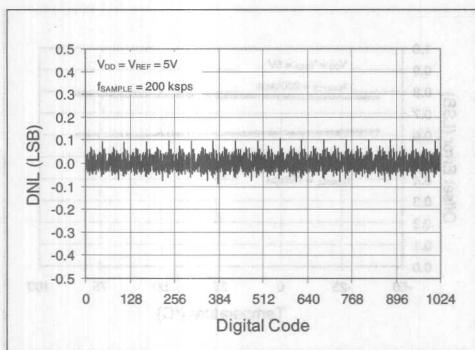


FIGURE 2-13: Differential Nonlinearity (DNL) vs. Code (Representative Part).

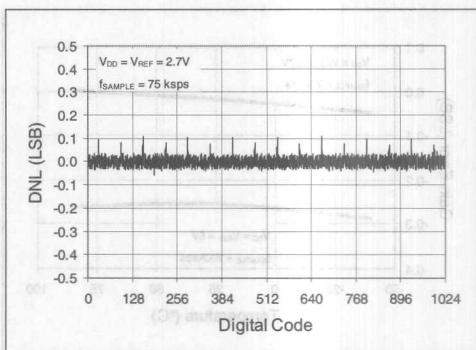


FIGURE 2-16: Differential Nonlinearity (DNL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

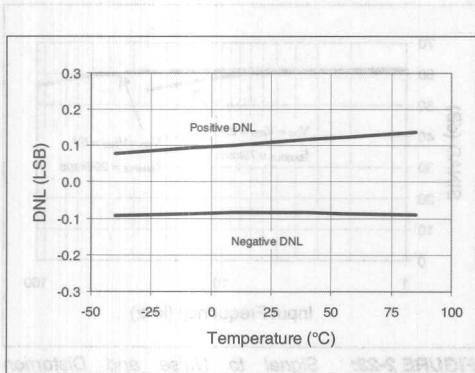


FIGURE 2-14: Differential Nonlinearity (DNL) vs. Temperature.

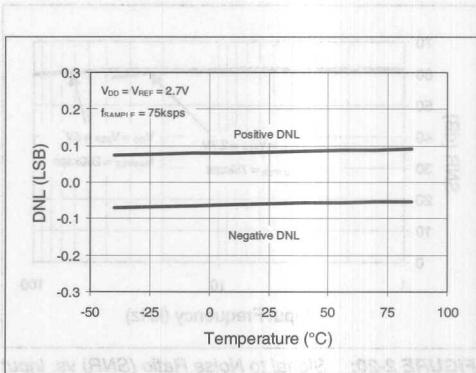


FIGURE 2-17: Differential Nonlinearity (DNL) vs. Temperature ($V_{DD} = 2.7V$).

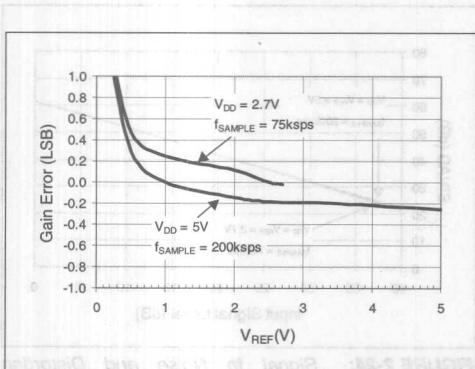


FIGURE 2-15: Gain Error vs. V_{REF} .

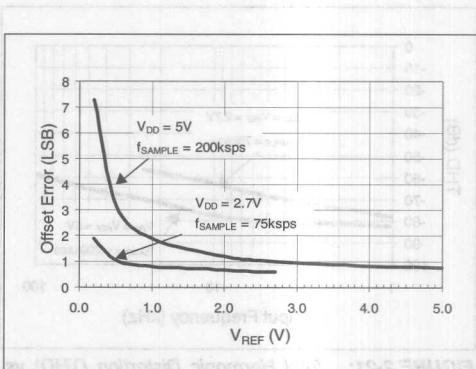


FIGURE 2-18: Offset Error vs. V_{REF} .

MCP3001

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{kspS}$, $f_{CLK} = 14 \times \text{Sample Rate}$, $T_A = 25^\circ\text{C}$

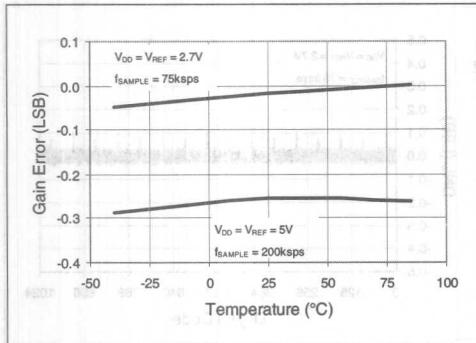


FIGURE 2-19: Gain Error vs. Temperature.

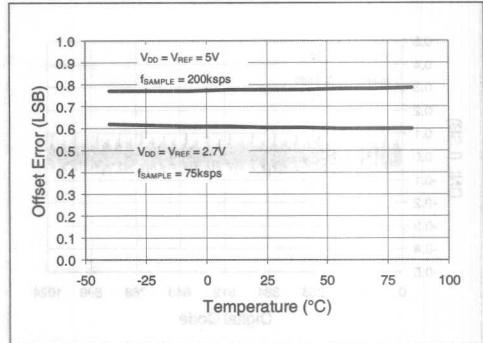


FIGURE 2-22: Offset Error vs. Temperature.

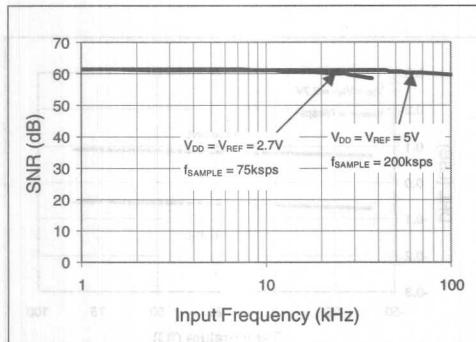


FIGURE 2-20: Signal to Noise Ratio (SNR) vs. Input Frequency.

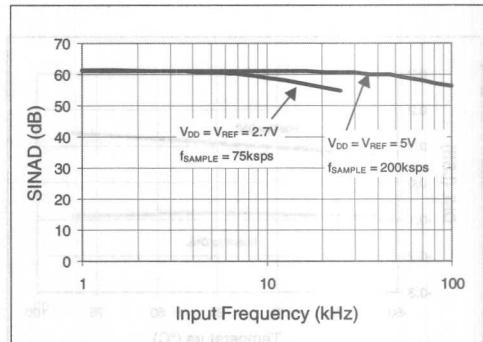


FIGURE 2-23: Signal to Noise and Distortion (SINAD) vs. Input Frequency.

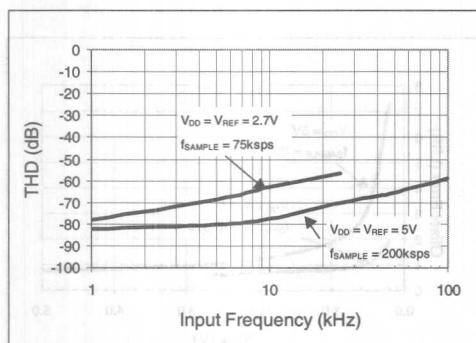


FIGURE 2-21: Total Harmonic Distortion (THD) vs. Input Frequency.

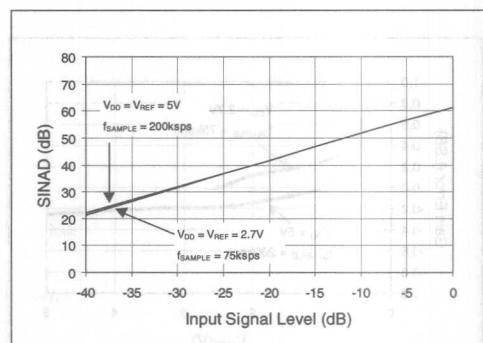


FIGURE 2-24: Signal to Noise and Distortion (SINAD) vs. Input Signal Level.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 14 \times \text{Sample Rate}$, $T_A = 25^\circ\text{C}$

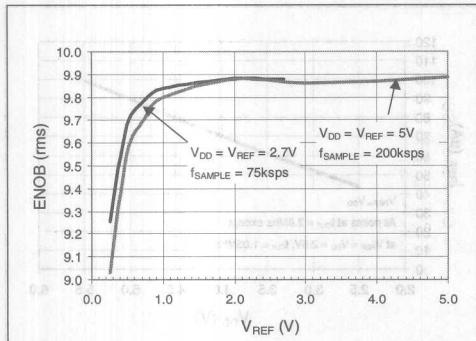


FIGURE 2-25: Effective Number of Bits (ENOB) vs. V_{REF} .

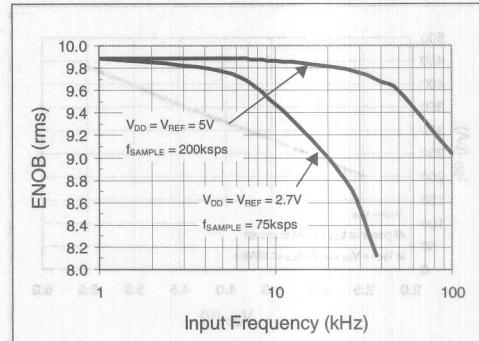


FIGURE 2-28: Effective Number of Bits (ENOB) vs. Input Frequency.

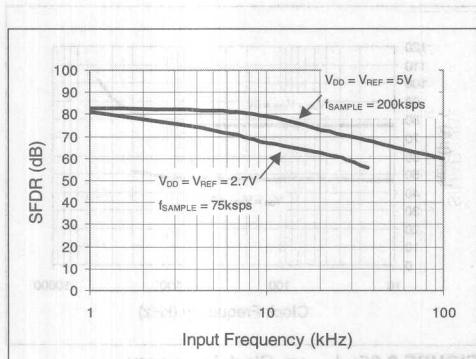


FIGURE 2-26: Spurious Free Dynamic Range (SFDR) vs. Input Frequency.

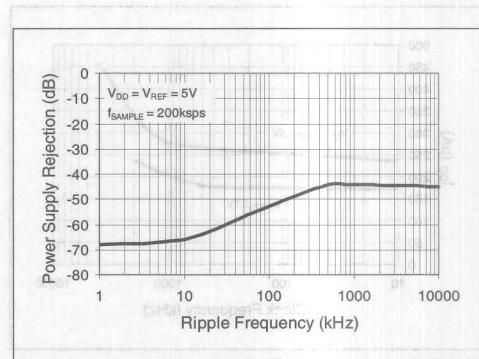


FIGURE 2-29: Power Supply Rejection (PSR) vs. Ripple Frequency.

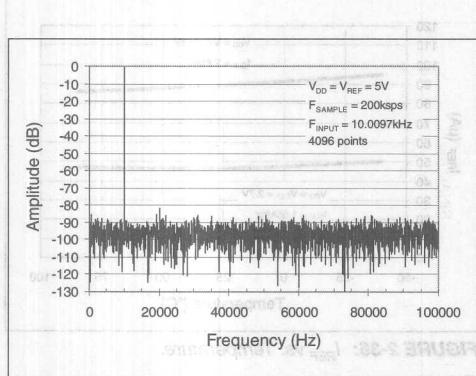


FIGURE 2-27: Frequency Spectrum of 10kHz input (Representative Part).

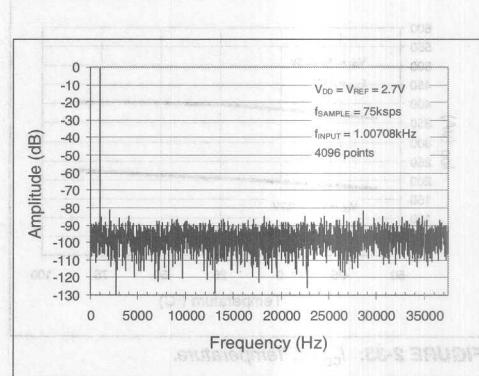


FIGURE 2-30: Frequency Spectrum of 1kHz input (Representative Part, $V_{DD} = 2.7V$).

MCP3001

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 14 * \text{Sample Rate}$, $T_A = 25^\circ\text{C}$

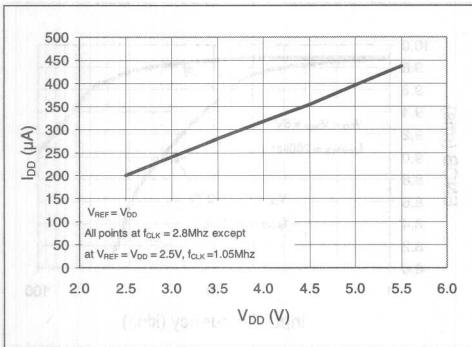


FIGURE 2-31: I_{DD} vs. V_{DD} .

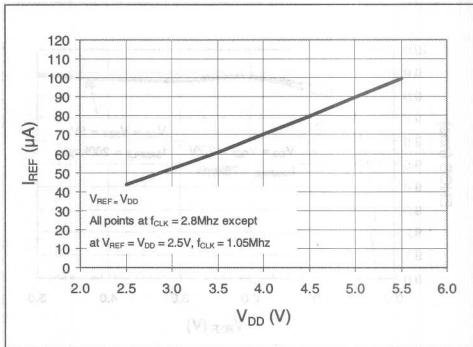


FIGURE 2-34: I_{REF} vs. V_{DD} .

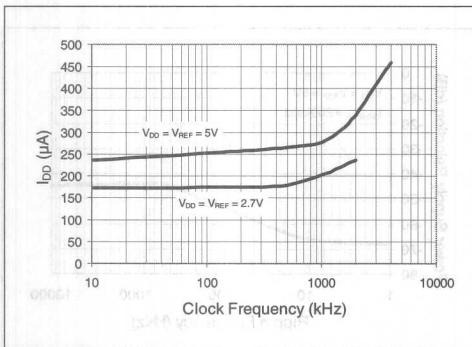


FIGURE 2-32: I_{DD} vs. Clock Frequency.

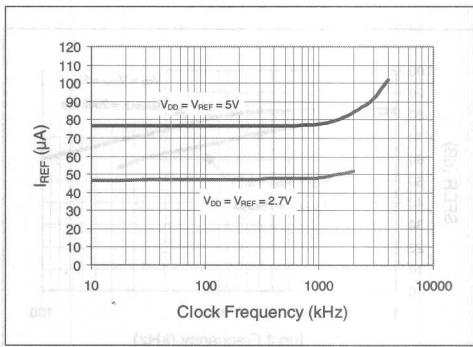


FIGURE 2-35: I_{REF} vs. Clock Frequency.

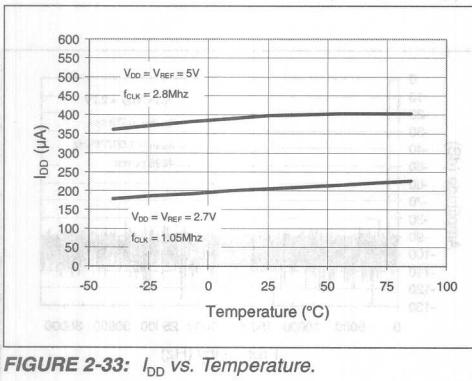


FIGURE 2-33: I_{DD} vs. Temperature.

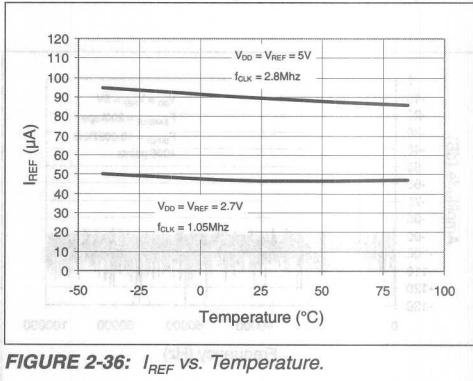


FIGURE 2-36: I_{REF} vs. Temperature.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 14^*\text{Sample Rate}$, $T_A = 25^\circ\text{C}$

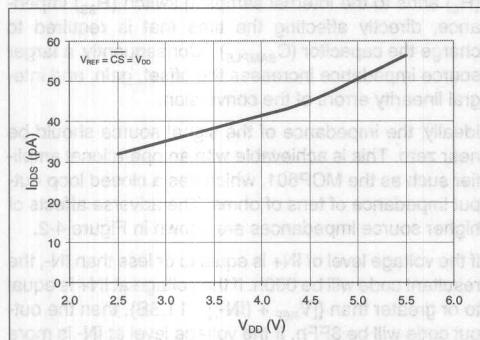


FIGURE 2-37: I_{DDS} vs. V_{DD} .

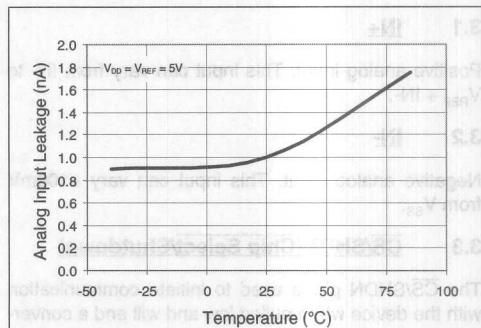


FIGURE 2-39: Analog Input Leakage Current vs. Temperature.

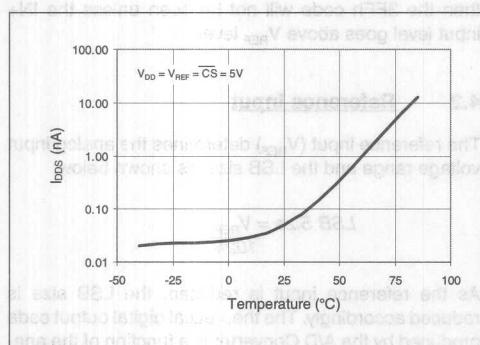


FIGURE 2-38: I_{DDS} vs. Temperature.

3

Datasheets

4.0 DETAILED OPERATION

The MCP3001 is a single-channel, 10-bit SAR ADC with a sample-and-hold function. It features a 10-bit resolution, a sampling rate of up to 200ksps, and a low power consumption of less than 100µW. The device is designed for use in portable applications where low power consumption is a key requirement. The MCP3001 is manufactured using a high-performance CMOS process and is available in a 16-pin TSSOP package.

The MCP3001 has a simple pin-out configuration, making it easy to interface with microcontrollers or other digital systems. The device features a single analog input pin, a digital output pin, and a clock input pin. The digital output pin provides the 10-bit digital representation of the analog input signal, while the clock input pin controls the sampling rate of the device.

The MCP3001 is a high-performance ADC that is ideal for a wide range of applications. Its low power consumption and high sampling rate make it suitable for use in portable devices such as mobile phones, PDAs, and MP3 players. The device's high resolution and accurate conversion results make it a popular choice for medical, scientific, and industrial applications.

MCP3001

3.0 PIN DESCRIPTIONS

3.1 IN+

Positive analog input. This input can vary from IN- to $V_{REF} + IN_-$.

3.2 IN-

Negative analog input. This input can vary $\pm 100mV$ from V_{SS} .

3.3 CS/SHDN(Chip Select/Shutdown)

The CS/SHDN pin is used to initiate communication with the device when pulled low and will end a conversion and put the device in low power standby when pulled high. The CS/SHDN pin must be pulled high between conversions.

3.4 CLK (Serial Clock)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 6.2 for constraints on clock speed.

3.5 DOUT (Serial Data output)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

4.0 DEVICE OPERATION

The MCP3001 A/D converter employs a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the first rising edge of the serial clock after CS has been pulled low. Following this sample time, the input switch of the converter opens and the device uses the collected charge on the internal sample and hold capacitor to produce a serial 10-bit digital output code. Conversion rates of 200ksps are possible on the MCP3001. See Section 6.2 for information on minimum clock rates. Communication with the device is done using a 3-wire SPI-compatible interface.

4.1 Analog Inputs

The MCP3001 provides a single pseudo-differential input. The IN+ input can range from IN- to $(V_{REF} + IN_-)$. The IN- input is limited to $\pm 100mV$ from the V_{SS} rail. The IN- input can be used to cancel small signal common-mode noise which is present on both the IN+ and IN- inputs.

For the A/D Converter to meet specification, the charge holding capacitor (C_{SAMPLE}) must be given enough time to acquire a 10-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

In this diagram, it is shown that the source impedance (R_s) adds to the internal sampling switch (R_{SS}) impedance, directly affecting the time that is required to charge the capacitor (C_{SAMPLE}). Consequently, a larger source impedance increases the offset, gain, and integral linearity errors of the conversion.

Ideally, the impedance of the signal source should be near zero. This is achievable with an operational amplifier such as the MCP601, which has a closed loop output impedance of tens of ohms. The adverse affects of higher source impedances are shown in Figure 4-2.

If the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h. If the voltage at IN+ is equal to or greater than $\{[V_{REF} + (IN_-)] - 1 \text{ LSB}\}$, then the output code will be 3FFh. If the voltage level at IN- is more than 1 LSB below V_{SS} , then the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS} , then the 3FFh code will not be seen unless the IN+ input level goes above V_{REF} level.

4.2 Reference Input

The reference input (V_{REF}) determines the analog input voltage range and the LSB size, as shown below.

$$\text{LSB Size} = \frac{V_{REF}}{1024}$$

As the reference input is reduced, the LSB size is reduced accordingly. The theoretical digital output code produced by the A/D Converter is a function of the analog input signal and the reference input as shown below.

$$\text{Digital Output Code} = \frac{1024 * V_{IN}}{V_{REF}}$$

where:

$$V_{IN} = \text{analog input voltage} = V(IN+) - V(IN-)$$

$$V_{REF} = \text{reference voltage}$$

When using an external voltage reference device, the system designer should always refer to the manufacturer's recommendations for circuit layout. Any instability in the operation of the reference device will have a direct effect on the operation of the ADC.

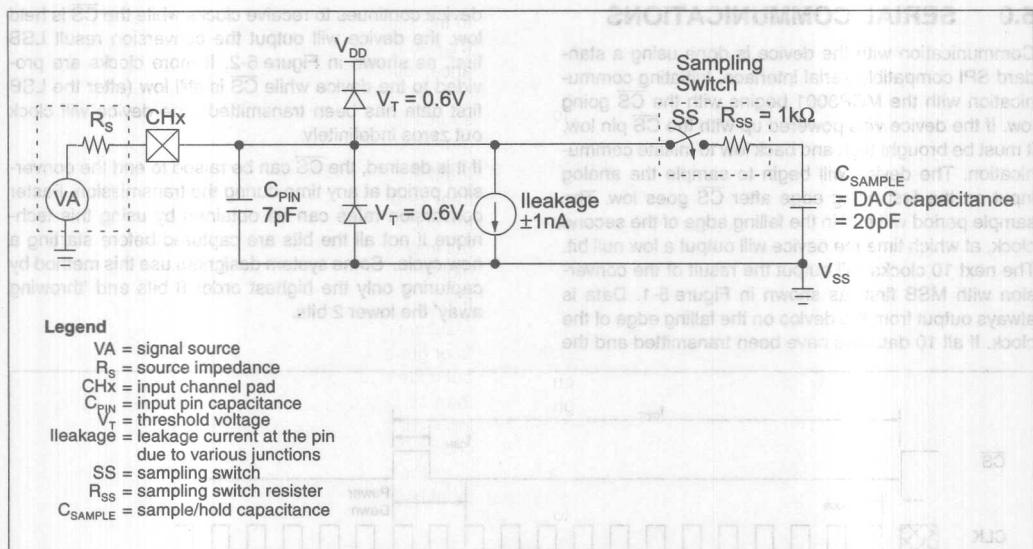
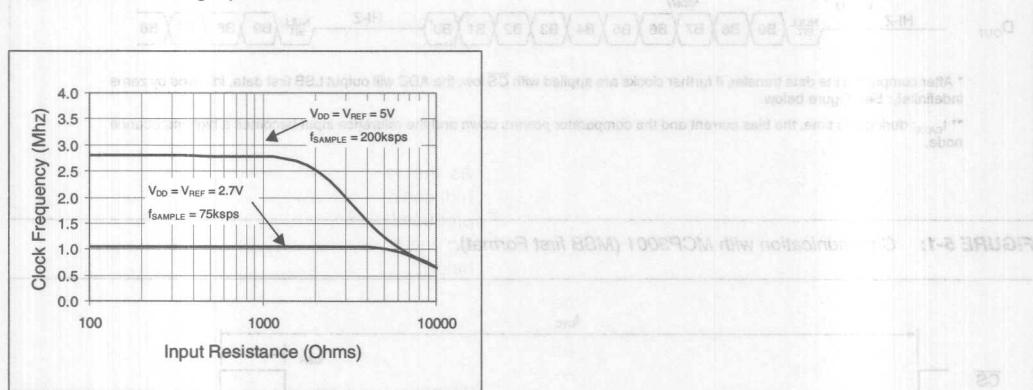
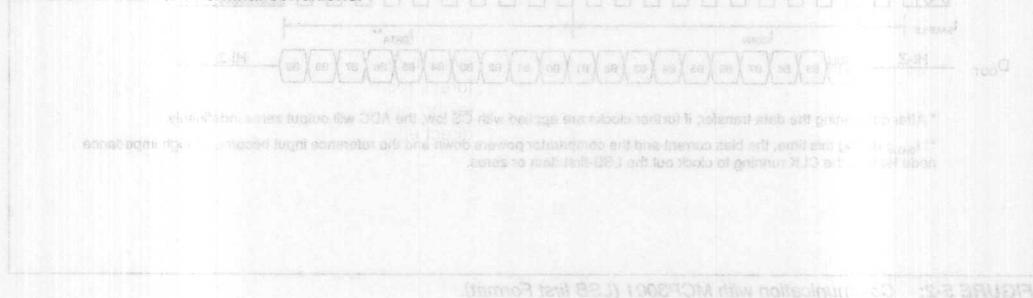


FIGURE 4-1: Analog Input Model.

FIGURE 4-2: Maximum Clock Frequency vs. Input Resistance (R_S) to maintain less than a 0.1LSB deviation in INL from nominal conditions.

MCP3001

5.0 SERIAL COMMUNICATIONS

Communication with the device is done using a standard SPI compatible serial interface. Initiating communication with the MCP3001 begins with the \overline{CS} going low. If the device was powered up with the \overline{CS} pin low, it must be brought high and back low to initiate communication. The device will begin to sample the analog input on the first rising edge after \overline{CS} goes low. The sample period will end in the falling edge of the second clock, at which time the device will output a low null bit. The next 10 clocks will output the result of the conversion with MSB first, as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 10 data bits have been transmitted and the

device continues to receive clocks while the \overline{CS} is held low, the device will output the conversion result LSB first, as shown in Figure 5-2. If more clocks are provided to the device while \overline{CS} is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If it is desired, the \overline{CS} can be raised to end the conversion period at any time during the transmission. Faster conversion rates can be obtained by using this technique if not all the bits are captured before starting a new cycle. Some system designers use this method by capturing only the highest order 8 bits and ‘throwing away’ the lower 2 bits.

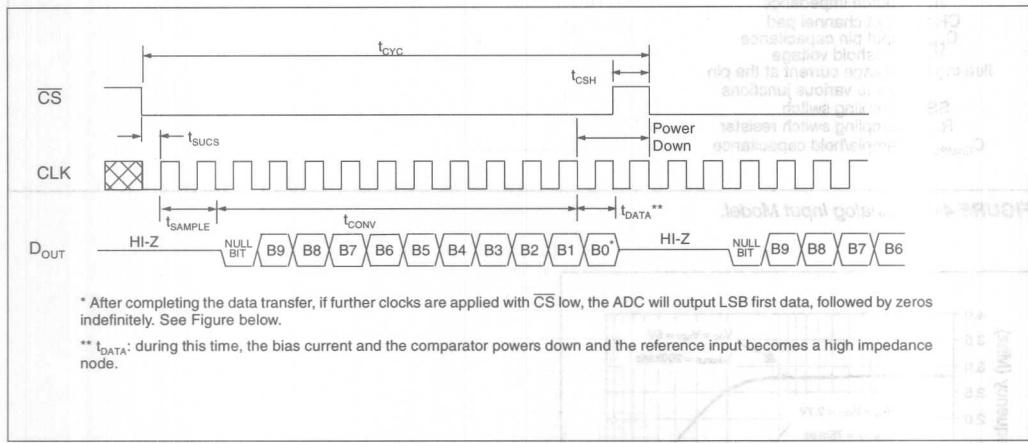


FIGURE 5-1: Communication with MCP3001 (MSB first Format).

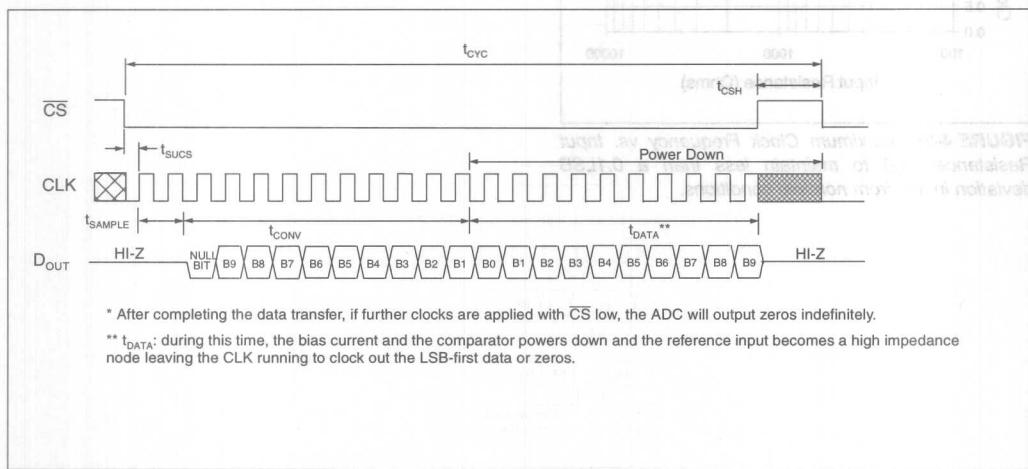


FIGURE 5-2: Communication with MCP3001 (LSB first Format).

6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3001 with Microcontroller SPI Ports

With most microcontroller SPI ports, it is required to clock out eight bits at a time. If this is the case, it will be necessary to provide more clocks than are required for the MCP3001. As an example, Figure 6-1 and Figure 6-2 show how the MCP3001 can be interfaced to a microcontroller with a standard SPI port. Since the MCP3001 always clocks data out on the falling edge of clock, the MCU SPI port must be configured to match this operation. SPI Mode 0,0 (clock idles low) and SPI Mode 1,1 (clock idles high) are both compatible with the MCP3001. Figure 6-1 depicts the operation shown in SPI Mode 0,0, which requires that the CLK from the microcontroller idles in the 'low' state. As shown in the diagram, the MSB is clocked out of the ADC on the falling edge of the third clock pulse. After the first eight clocks have been sent to the device, the microcontrol-

ler's receive buffer will contain two unknown bits (the output is at high impedance for the first two clocks), the null bit and the highest order five bits of the conversion. After the second eight clocks have been sent to the device, the MCU receive register will contain the lowest order five bits and the B1-B4 bits repeated as the ADC has begun to shift out LSB first data with the extra clocks. Typical procedure would then call for the lower order byte of data to be shifted right by three bits to remove the extra B1-B4 bits. The B9-B5 bits are then rotated 3 bits to the right with B7-B5 rotating from the high order byte to the lower order byte. Easier manipulation of the converted data can be obtained by using this method.

Figure 6-2 shows SPI Mode 1,1 communication which requires that the clock idles in the high state. As with mode 0,0, the ADC outputs data on the falling edge of the clock and the MCU latches data from the ADC in on the rising edge of the clock.

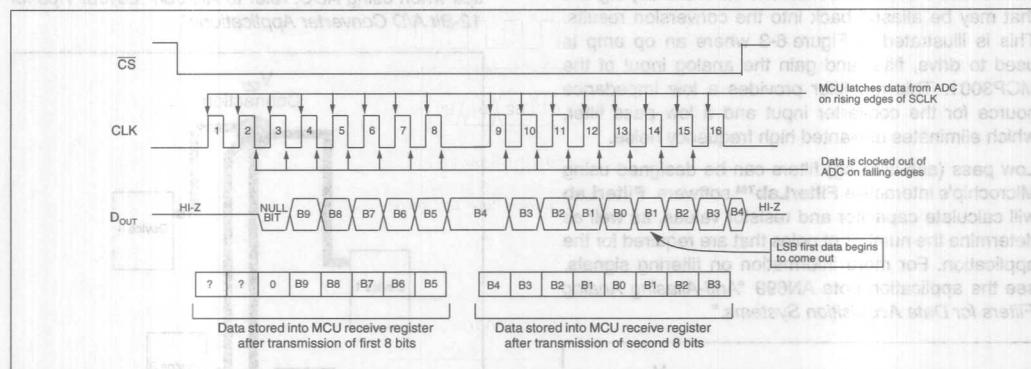


FIGURE 6-1: SPI Communication with the MCP3001 using 8 bit segments (Mode 0,0: SCLK idles low).

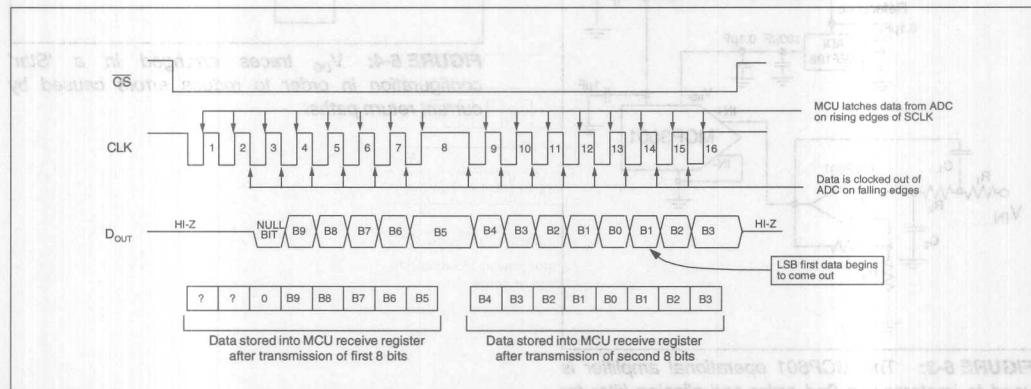


FIGURE 6-2: SPI Communication with the MCP3001 using 8 bit segments (Mode 1,1: SCLK idles high).

6.2 Maintaining Minimum Clock Speed

When the MCP3001 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample cap while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample cap for 700 μ s at $V_{DD} = 2.7V$ and 1.5ms at $V_{DD} = 5V$. This means that at $V_{DD} = 2.7V$, the time it takes to transmit the first 14 clocks must not exceed 700 μ s. Failure to meet this criterion may induce linearity errors into the conversion outside the rated specifications.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the ADC is not a low impedance source, it will have to be buffered or inaccurate conversion results may occur. See Figure 4-2. It is also recommended that a filter be used to eliminate any signals that may be aliased back into the conversion results. This is illustrated in Figure 6-3 where an op amp is used to drive, filter and gain the analog input of the MCP3001. This amplifier provides a low impedance source for the converter input and a low pass filter, which eliminates unwanted high frequency noise.

Low pass (anti-aliasing) filters can be designed using Microchip's interactive **FilterLab™** software. **FilterLab** will calculate capacitor and resistor values, as well as determine the number of poles that are required for the application. For more information on filtering signals, see the application note AN699 "Anti-Aliasing Analog Filters for Data Acquisition Systems."

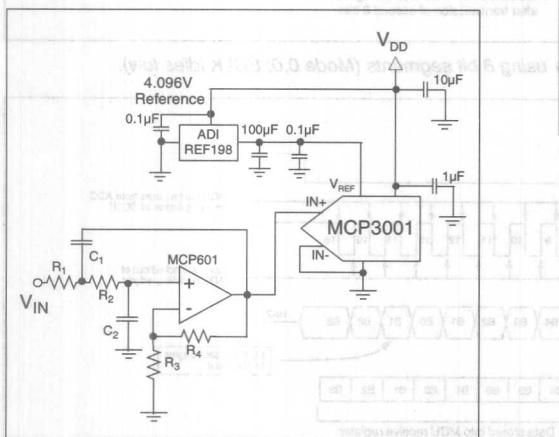


FIGURE 6-3: The MCP601 operational amplifier is used to implement a 2nd order anti-aliasing filter for the signal being converted by the MCP3001.

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of 1 μ F is recommended.

Digital and analog traces should be separated as much as possible on the board and no traces should run underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating current return paths and associated errors. See Figure 6-4. For more information on layout tips when using ADC, refer to AN-688 "Layout Tips for 12-Bit A/D Converter Applications".

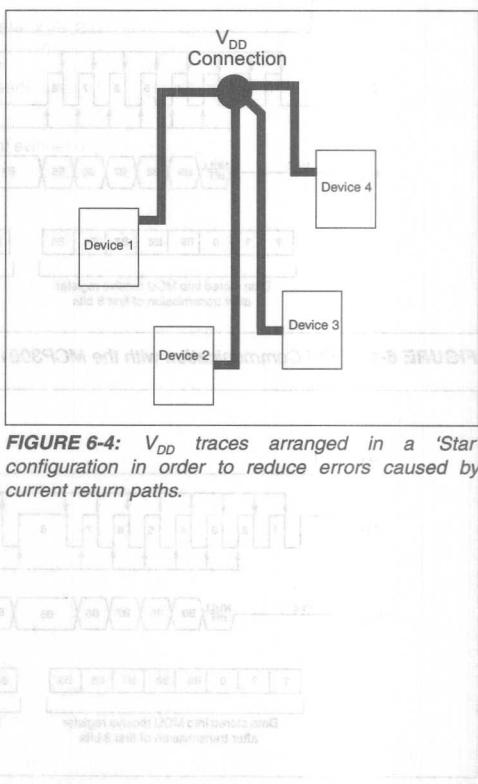


FIGURE 6-4: V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.

MCP3001 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

MCP3001 - X /X	
Package:	P = PDIP (8 lead) SN = SOIC (150 mil Body), 8 lead ST = TSSOP, 8 lead
Temperature Range:	I = -40°C to +85°C
Device:	MCP3001 = 10-Bit Serial A/D Converter MCP3001T = 10-Bit Serial A/D Converter on tape and reel (SOIC and TSSOP packages only)

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCP3001

NOTES:



MICROCHIP

MCP3002

2.7V Dual Channel 10-Bit A/D Converter with SPI™ Serial Interface

FEATURES

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI® serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200ksps max sampling rate at $V_{DD} = 5V$
- 75ksps max sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
 - 5nA typical standby current, 2 μ A max
 - 550 μ A max. active current at 5V
- Industrial temp range: -40°C to +85°C
- 8-pin PDIP SOIC and TSSOP packages

APPLICATIONS

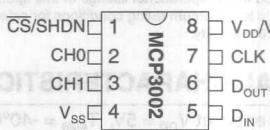
- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

DESCRIPTION

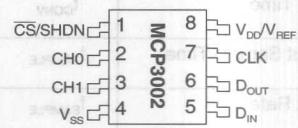
The Microchip Technology Inc. MCP3002 is a successive approximation 10-bit Analog-to-Digital (A/D) Converter with on-board sample and hold circuitry. The MCP3002 is programmable to provide a single pseudo-differential input pair or dual single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are both specified at ± 1 LSB. Communication with the device is done using a simple serial interface compatible with the SPI protocol. The device is capable of conversion rates of up to 200ksps at 5V and 75ksps at 2.7V. The MCP3002 device operates over a broad voltage range (2.7V - 5.5V). Low current design permits operation with a typical standby current of 5nA and a typical active current of 375 μ A. The MCP3002 is offered in 8-pin PDIP, TSSOP and 150mil SOIC packages.

PACKAGE TYPES

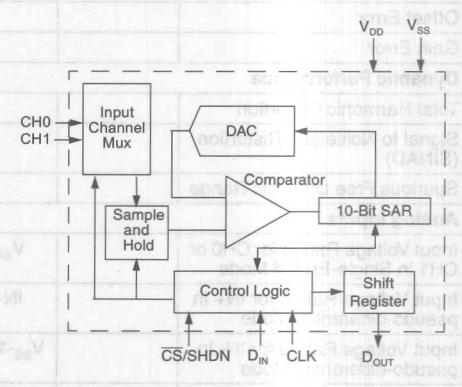
PDIP



SOIC, TSSOP



FUNCTIONAL BLOCK DIAGRAM



3

Datasheets

SPI is a trademark of Motorola Inc.

MCP3002

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to V_{DD} +0.6V
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	> 4kV

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
V_{DD}/V_{REF}	+2.7V To 5.5V Power Supply and Reference Voltage Input
CH0	Channel 0 Analog Input
CH1	Channel 1 Analog Input
CLK	Serial Clock
D _{IN}	Serial Data In
D _{OUT}	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input

ELECTRICAL CHARACTERISTICS

All parameters apply at $V_{DD} = 5V$, $T_{AMB} = -40^\circ C$ to $+85^\circ C$, $f_{SAMPLE} = 200\text{ksps}$ and $f_{CLK} = 16f_{SAMPLE}$ unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_{AMB} = 25^\circ C$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Conversion Rate						
Conversion Time	t_{CONV}			10	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}			200 75	ksps ksps	$V_{DD} = 5V$ $V_{DD} = 2.7V$
DC Accuracy						
Resolution			10		bits	
Integral Nonlinearity	INL		±0.5	±1	LSB	
Differential Nonlinearity	DNL		±0.25	±1	LSB	No missing codes over temperature
Offset Error				±1.5	LSB	
Gain Error				±1	LSB	
Dynamic Performance						
Total Harmonic Distortion			-76		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Signal to Noise and Distortion (SINAD)			61		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Spurious Free Dynamic Range			78		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Analog Inputs						
Input Voltage Range for CH0 or CH1 in Single-Ended Mode		V_{SS}		V_{DD}	V	
Input Voltage Range for IN+ In pseudo-differential Mode		IN-		$V_{DD} + IN-$		
Input Voltage Range for IN- In pseudo-differential Mode		$V_{SS} - 100$		$V_{SS} + 100$	mV	
Leakage Current			0.001	±1	μA	
Switch Resistance	R_{SS}		1K		Ω	See Figure 4-1
Sample Capacitor	C_{SAMPLE}		20		pF	See Figure 4-1

ELECTRICAL CHARACTERISTICS (CONTINUED)

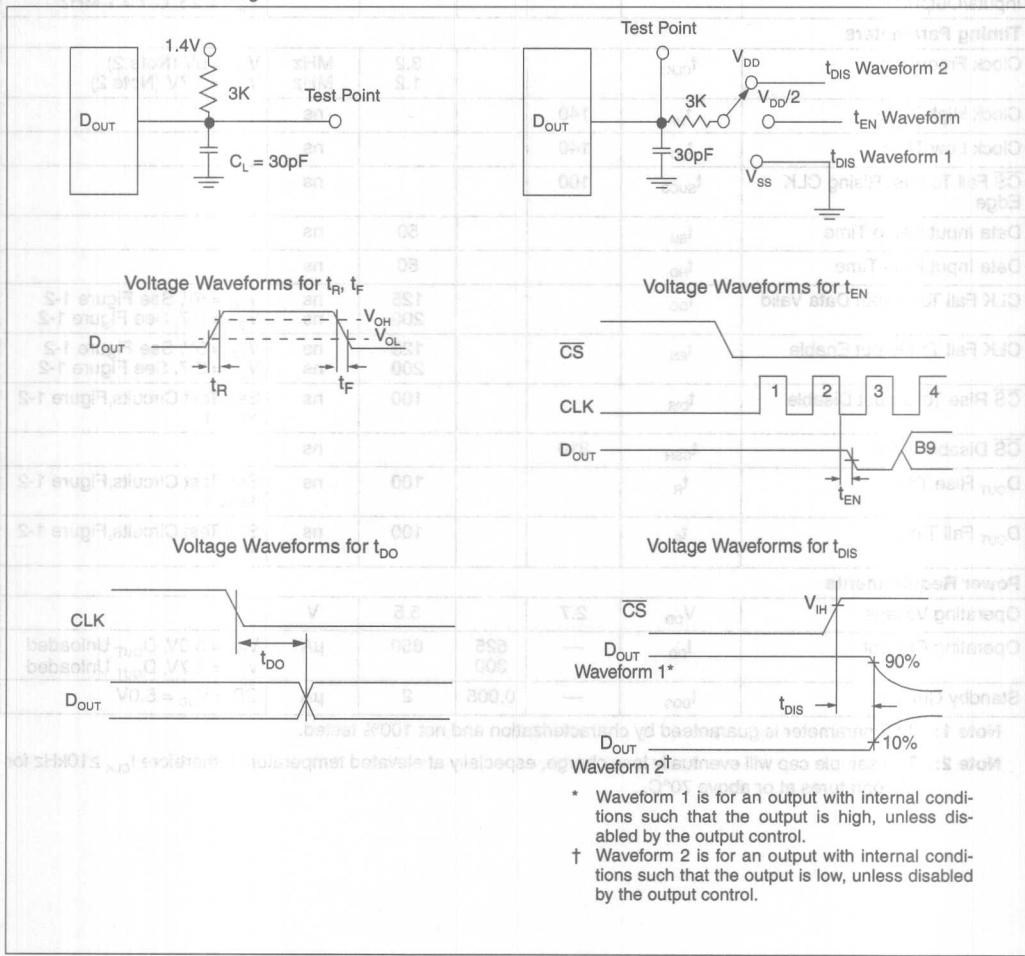
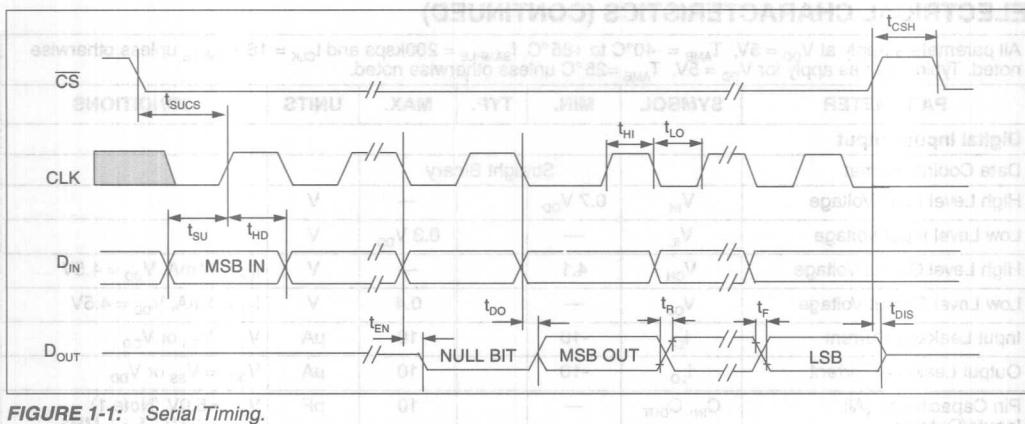
All parameters apply at $V_{DD} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 200\text{ksps}$ and $f_{CLK} = 16 \times f_{SAMPLE}$ unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_{AMB} = 25^{\circ}C$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Digital Input/Output						
Data Coding Format						
High Level Input Voltage	V_{IH}	$0.7 V_{DD}$	—	—	V	
Low Level Input Voltage	V_{IL}	—	—	$0.3 V_{DD}$	V	
High Level Output Voltage	V_{OH}	4.1	—	—	V	$I_{OH} = -1\text{mA}$, $V_{DD} = 4.5V$
Low Level Output Voltage	V_{OL}	—	—	0.4	V	$I_{OL} = 1\text{mA}$, $V_{DD} = 4.5V$
Input Leakage Current	I_{LI}	-10	—	10	μA	$V_{IN} = V_{SS}$ or V_{DD}
Output Leakage Current	I_{LO}	-10	—	10	μA	$V_{OUT} = V_{SS}$ or V_{DD}
Pin Capacitance (All Inputs/Outputs)	C_{IN}, C_{OUT}	—	—	10	pF	$V_{DD} = 5.0V$ (Note 1) $T_{AMB} = 25^{\circ}C$, $f = 1\text{ MHz}$
Timing Parameters						
Clock Frequency	f_{CLK}	—	—	3.2 1.2	MHz MHz	$V_{DD} = 5V$ (Note 2) $V_{DD} = 2.7V$ (Note 2)
Clock High Time	t_{HI}	140	—	—	ns	
Clock Low Time	t_{LO}	140	—	—	ns	
CS Fall To First Rising CLK Edge	t_{SUCS}	100	—	—	ns	
Data Input Setup Time	t_{SU}	—	—	50	ns	
Data Input Hold Time	t_{HD}	—	—	50	ns	
CLK Fall To Output Data Valid	t_{DO}	—	—	125 200	ns ns	$V_{DD} = 5V$, See Figure 1-2 $V_{DD} = 2.7V$, See Figure 1-2
CLK Fall To Output Enable	t_{EN}	—	—	125 200	ns ns	$V_{DD} = 5V$, See Figure 1-2 $V_{DD} = 2.7V$, See Figure 1-2
CS Rise To Output Disable	t_{DIS}	—	—	100	ns	See Test Circuits, Figure 1-2 Note 1
CS Disable Time	t_{CSH}	310	—	—	ns	
D_{OUT} Rise Time	t_R	—	—	100	ns	See Test Circuits, Figure 1-2 Note 1
D_{OUT} Fall Time	t_F	—	—	100	ns	See Test Circuits, Figure 1-2 Note 1
Power Requirements						
Operating Voltage	V_{DD}	2.7	—	5.5	V	
Operating Current	I_{DD}	—	525 300	650	μA	$V_{DD} = 5.0V$, D_{OUT} Unloaded $V_{DD} = 2.7V$, D_{OUT} Unloaded
Standby Current	I_{BDS}	—	0.005	2	μA	$CS = V_{DD} = 5.0V$

Note 1: This parameter is guaranteed by characterization and not 100% tested.

Note 2: The sample cap will eventually lose charge, especially at elevated temperatures, therefore $f_{CLK} \geq 10\text{kHz}$ for temperatures at or above $70^{\circ}C$.

MCP3002



2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note: Unless otherwise indicated, $V_{DD} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 16 * f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

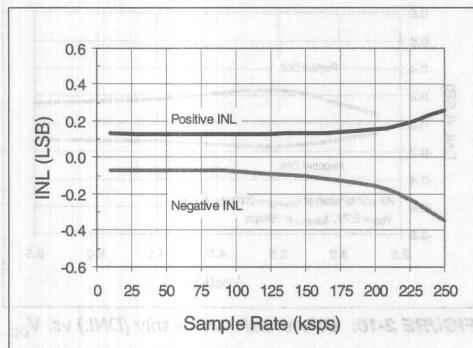


FIGURE 2-1: Integral Nonlinearity (INL) vs. Sample Rate.

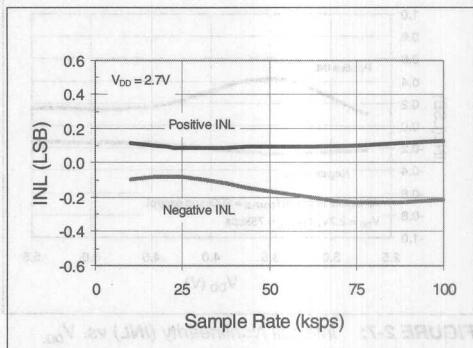


FIGURE 2-4: Integral Nonlinearity (INL) vs. Sample Rate ($V_{DD} = 2.7V$).

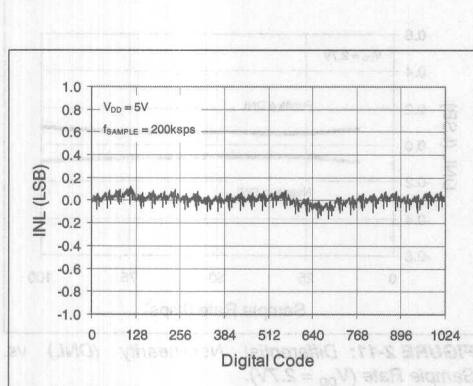


FIGURE 2-2: Integral Nonlinearity (INL) vs. Code.

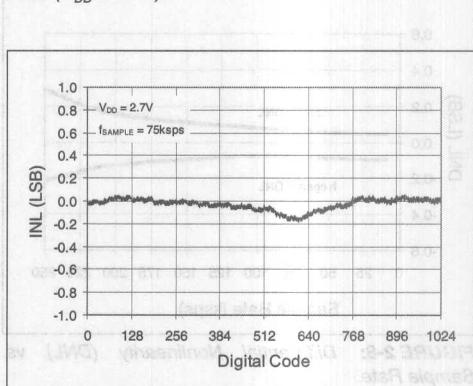


FIGURE 2-5: Integral Nonlinearity (INL) vs. Code ($V_{DD} = 2.7V$).

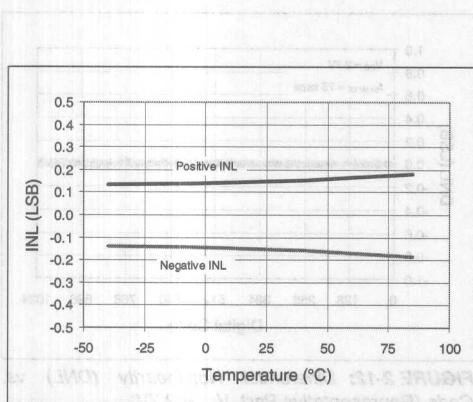


FIGURE 2-3: Integral Nonlinearity (INL) vs. Temperature.

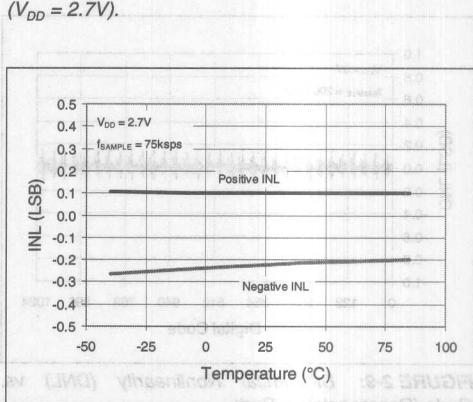


FIGURE 2-6: Integral Nonlinearity (INL) vs. Temperature ($V_{DD} = 2.7V$).

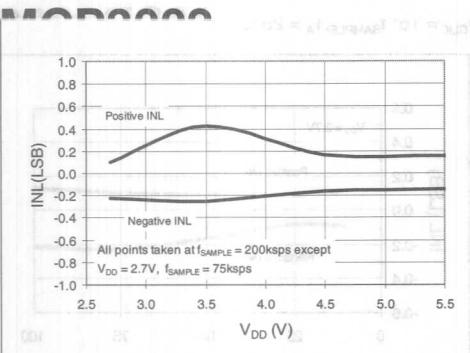


FIGURE 2-7: Integral Nonlinearity (INL) vs. V_{DD} .

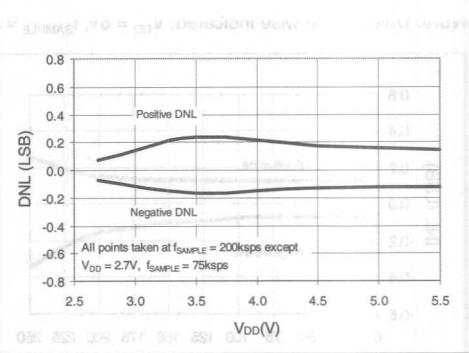


FIGURE 2-10: Differential Nonlinearity (DNL) vs. V_{DD} .

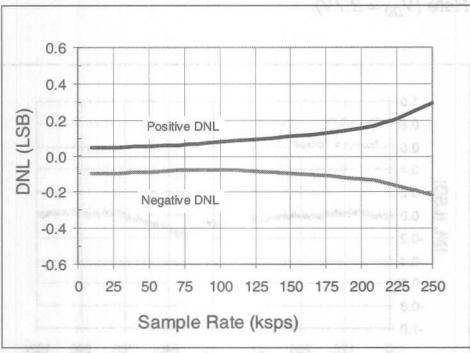


FIGURE 2-8: Differential Nonlinearity (DNL) vs. Sample Rate.

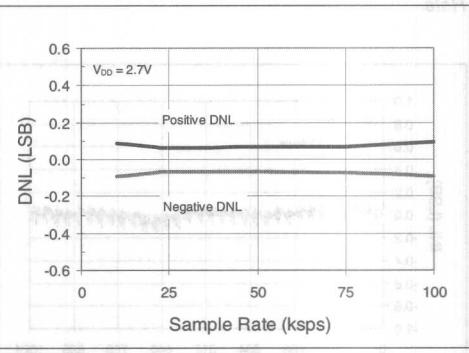


FIGURE 2-11: Differential Nonlinearity (DNL) vs. Sample Rate ($V_{DD} = 2.7V$).

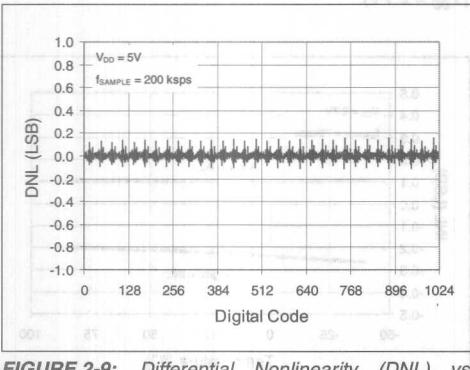


FIGURE 2-9: Differential Nonlinearity (DNL) vs. Code (Representative Part).

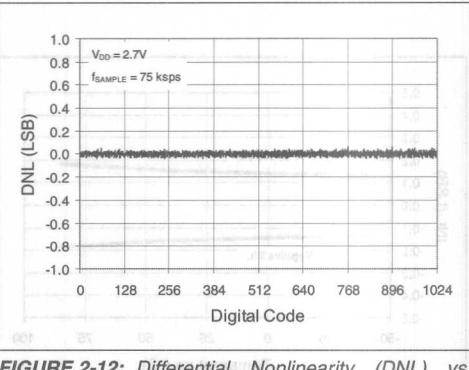


FIGURE 2-12: Differential Nonlinearity (DNL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

Note: Unless otherwise indicated, $V_{DD} = 5V$, $f_{SAMPLE} = 200\text{kspS}$, $f_{CLK} = 16 * f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

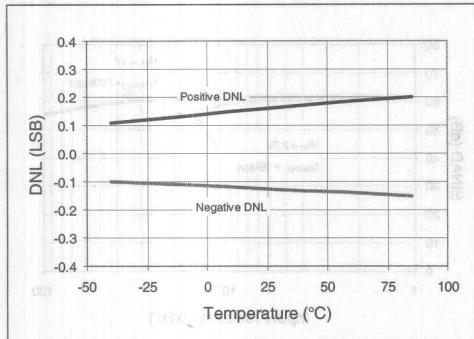


FIGURE 2-13: Differential Nonlinearity (DNL) vs. Temperature.

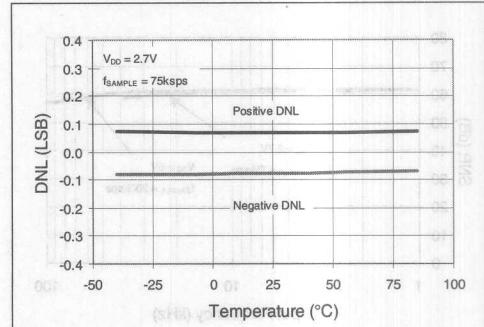


FIGURE 2-16: Differential Nonlinearity (DNL) vs. Temperature ($V_{DD} = 2.7V$).

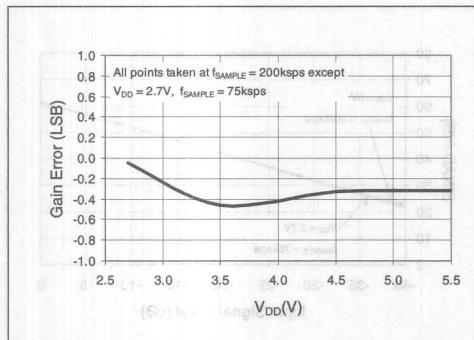


FIGURE 2-14: Gain Error vs. V_{DD} .

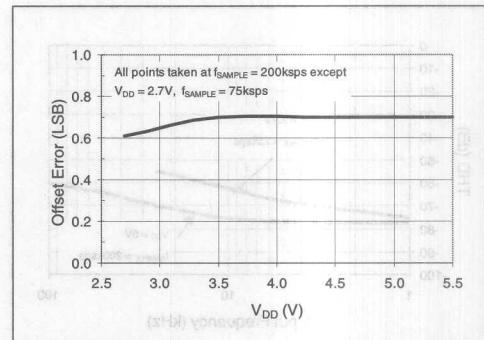


FIGURE 2-17: Offset Error vs. V_{DD} .

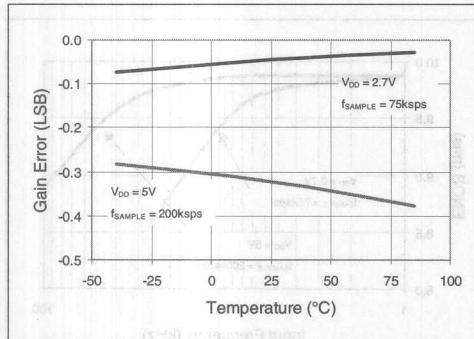


FIGURE 2-15: Gain Error vs. Temperature.

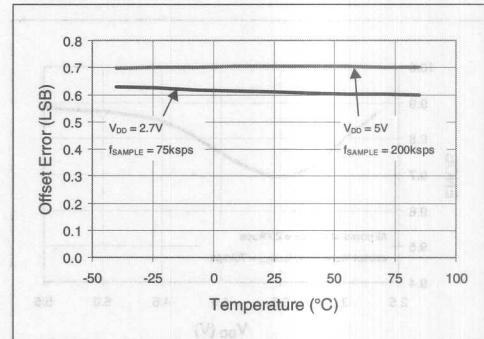


FIGURE 2-18: Offset Error vs. Temperature.

MCP3002

Note: Unless otherwise indicated, $V_{DD} = 5V$, $f_{SAMPLE} = 200\text{kspS}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

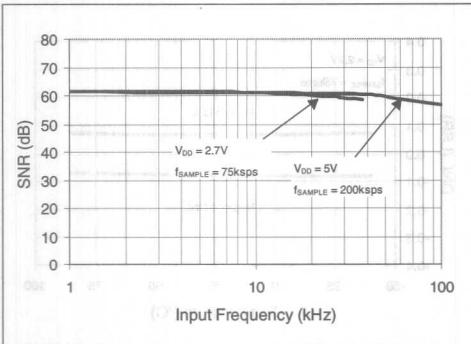


FIGURE 2-19: Signal to Noise Ratio (SNR) vs. Input Frequency.

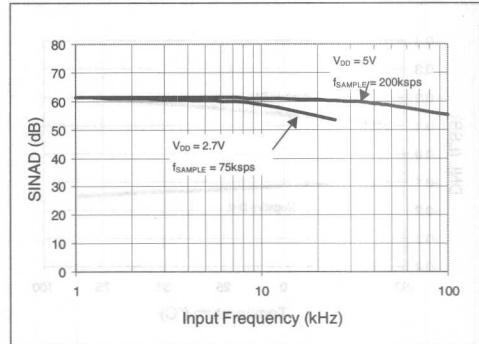


FIGURE 2-22: Signal to Noise and Distortion (SINAD) vs. Input Frequency.

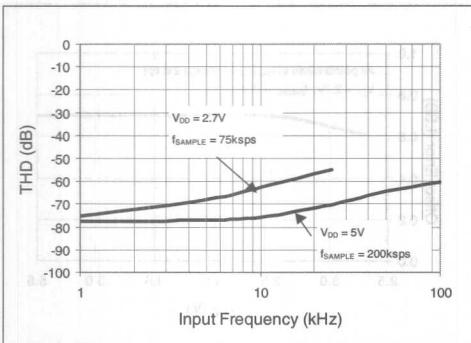


FIGURE 2-20: Total Harmonic Distortion (THD) vs. Input Frequency.

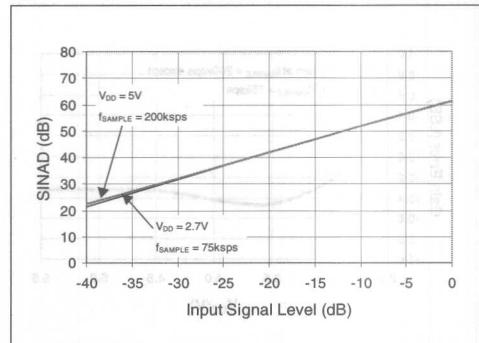


FIGURE 2-23: Signal to Noise and Distortion (SINAD) vs. Signal Level.

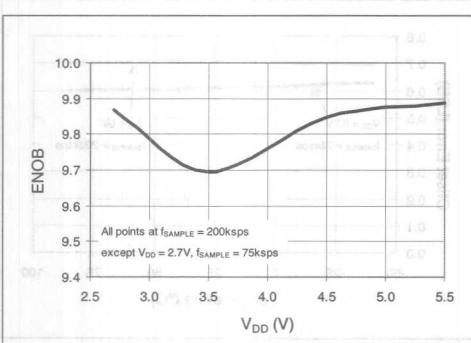


FIGURE 2-21: Effective number of bits (ENOB) vs. V_{DD} .

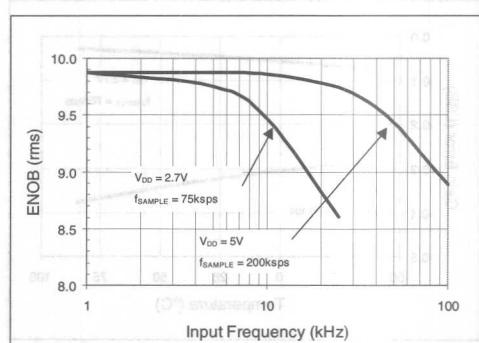


FIGURE 2-24: Effective Number of Bits (ENOB) vs. Input Frequency.

Note: Unless otherwise indicated, $V_{DD} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

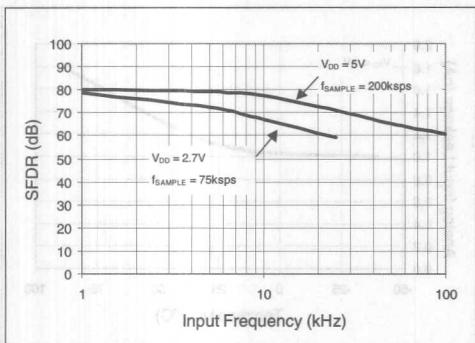


FIGURE 2-25: Spurious Free Dynamic Range (SFDR) vs. Input Frequency.

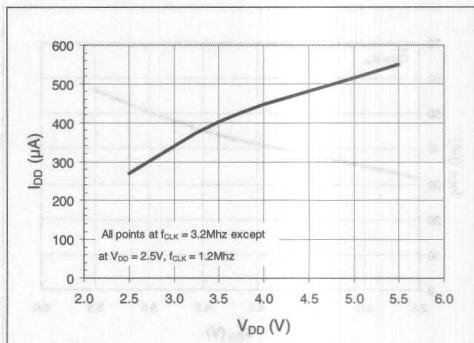


FIGURE 2-28: I_{DD} vs. V_{DD} .

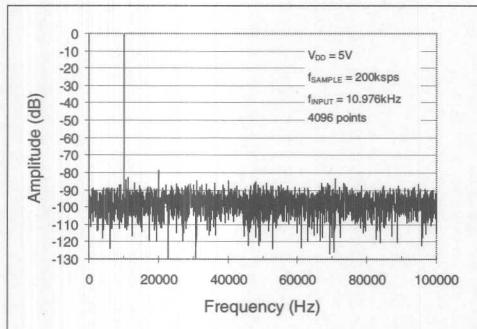


FIGURE 2-26: Frequency Spectrum of 10kHz input (Representative Part).

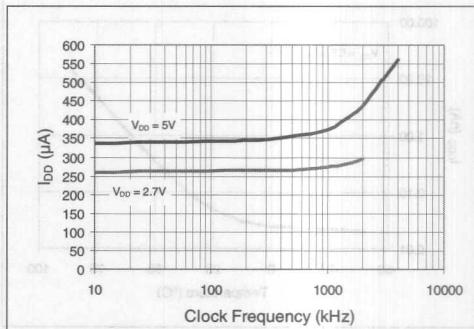


FIGURE 2-29: I_{DD} vs. Clock Frequency.

3

Datasheets

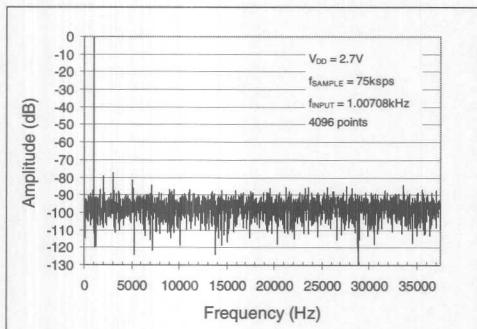


FIGURE 2-27: Frequency Spectrum of 1kHz input (Representative Part, $V_{DD} = 2.7V$).

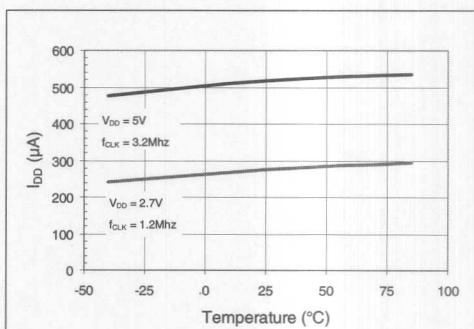


FIGURE 2-30: I_{DD} vs. Temperature.

MCP3002

Note: Unless otherwise indicated, $V_{DD} = 5V$, $f_{SAMPLE} = 200\text{ksps}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$ until otherwise specified.

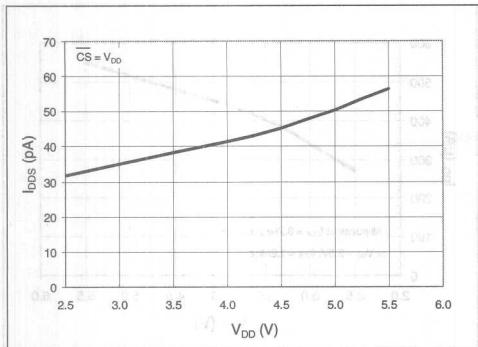


FIGURE 2-31: I_{DSS} vs. V_{DD} .

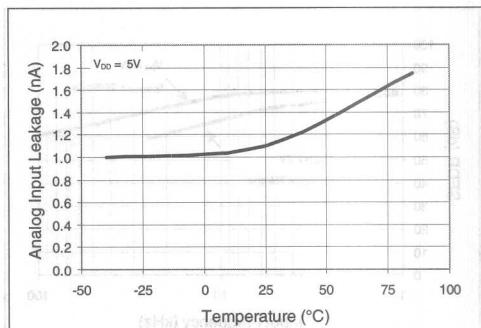


FIGURE 2-33: Analog input leakage current vs. Temperature.

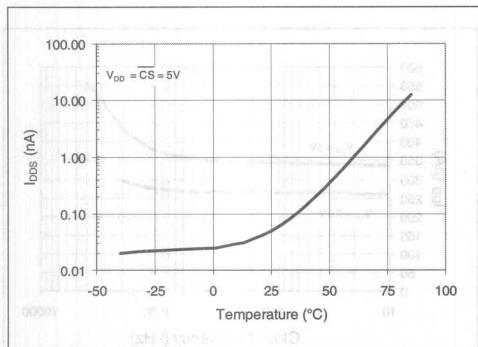


FIGURE 2-32: I_{DSS} vs. Temperature.

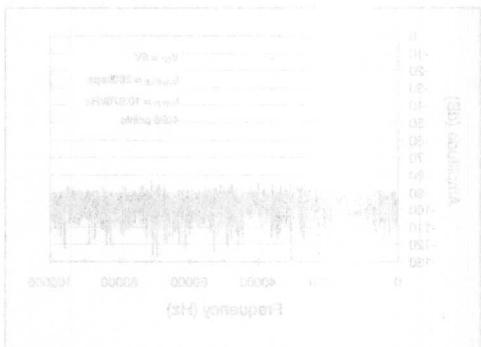
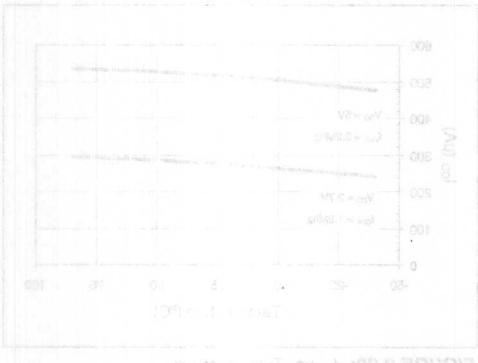


FIGURE 2-34: Frequency response vs. Frequency.



3.0 PIN DESCRIPTIONS

3.1 CH0/CH1

Analog inputs for channels 0 and 1 respectively. These channels can be programmed to be used as two independent channels in single ended-mode or as a single pseudo-differential input where one channel is IN+ and one channel is IN-. See Section 5.0 for information on programming the channel configuration.

3.2 CS/SHDN(Chip Select/Shutdown)

The CS/SHDN pin is used to initiate communication with the device when pulled low and will end a conversion and put the device in low power standby when pulled high. The CS/SHDN pin must be pulled high between conversions.

3.3 CLK (Serial Clock)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 6.2 for constraints on clock speed.

3.4 DIN (Serial Data Input)

The SPI port serial data input pin is used to clock in input channel configuration data.

3.5 DOUT (Serial Data output)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

4.0 DEVICE OPERATION

The MCP3002 A/D converter employs a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the second rising edge of the serial clock after the start bit has been received. Following this sample time, the input switch of the converter opens and the device uses the collected charge on the internal sample and hold capacitor to produce a serial 10-bit digital output code. Conversion rates of 200ksps are possible on the MCP3002. See Section 6.2 for information on minimum clock rates. Communication with the device is done using a 3-wire SPI compatible interface.

4.1 Analog Inputs

The MCP3002 device offers the choice of using the analog input channels configured as two single-ended inputs that are referenced to V_{SS} or a single pseudo-differential input. The configuration setup is done as part of the serial command before each conversion begins. When used in the pseudo-differential mode, CH0 and CH1 are programmed as the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ input can range from IN- to the reference voltage, V_{DD}. The IN- input is limited to ±100mV from the V_{SS} rail. The IN- input can be used to cancel small signal common-mode noise which is present on both the IN+ and IN- inputs.

For the A/D Converter to meet specification, the charge holding capacitor (C_{SAMPLE}) must be given enough time to acquire a 10-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

In this diagram, it is shown that the source impedance (R_S) adds to the internal sampling switch (R_{SS}) impedance, directly affecting the time that is required to charge the capacitor, C_{SAMPLE}. Consequently, larger source impedances increase the offset, gain, and integral linearity errors of the conversion.

Ideally, the impedance of the signal source should be near zero. This is achievable with an operational amplifier such as the MCP601 which has a closed loop output impedance of tens of ohms. The adverse effects of higher source impedances are shown in Figure 4-2.

When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h. If the voltage at IN+ is equal to or greater than {[V_{DD} + (IN-)] - 1 LSB}, then the output code will be 3FFh. If the voltage level at IN- is more than 1 LSB below V_{SS}, then the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS}, then the 3FFh code will not be seen unless the IN+ input level goes above V_{DD} level. If the voltage at IN+ is equal to or greater than {[V_{DD} + (IN-)] - 1 LSB}, then the output code will be 3FFh.

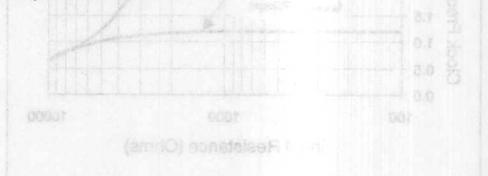


FIGURE 4-2: MC3002 Device Chip Response to Input Response. The graph shows the digital output response over a range of input voltages. The input voltage (IN+) is plotted on the x-axis, ranging from 0.0 to 1.0 Volts. The digital output (DOUT) is plotted on the y-axis, ranging from 0.0 to 0.8 Digital. The output shows a sawtooth-like pattern. It remains at 000h until approximately 0.5V, then transitions to 3FFh. It returns to 000h at approximately 0.0V and transitions back to 3FFh at approximately 0.5V. The period of the sawtooth is 1.5 clock cycles.

MCP3002

4.2 Digital Output Code

The digital output code produced by an A/D Converter is a function of the input signal and the reference voltage. For the MCP3002, V_{DD} is used as the reference voltage.

$$LSB\ Size = \frac{V_{REF}}{1024}$$

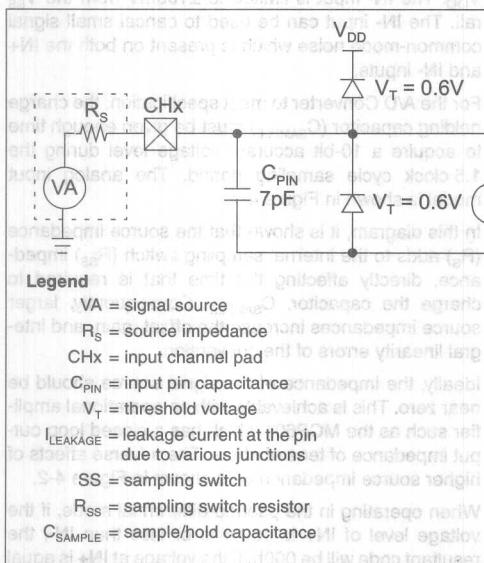


FIGURE 4-1: Analog Input Model.

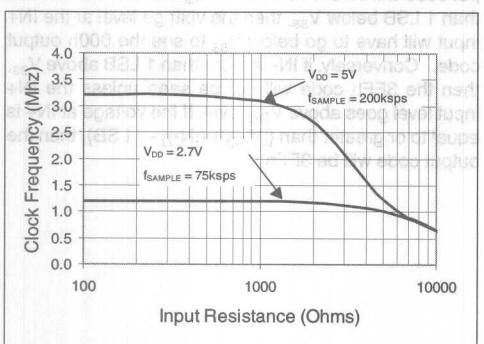
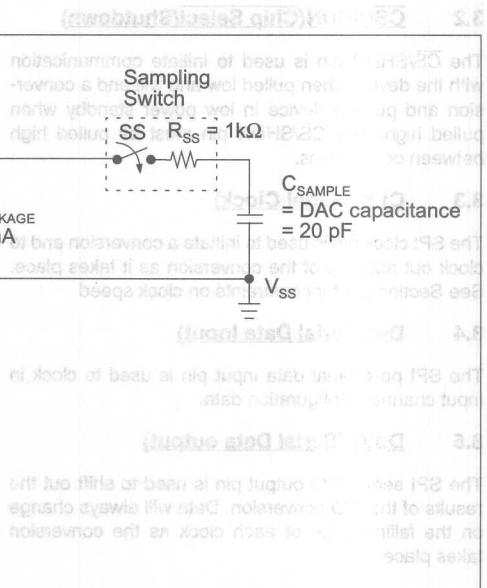


FIGURE 4-2: Maximum Clock Frequency vs. Input resistance (R_S) to maintain less than a 0.1LSB deviation in INL from nominal conditions.

As the V_{DD} level is reduced, the LSB size is reduced accordingly. The theoretical digital output code produced by the A/D Converter is shown below.

$$Digital\ Output\ Code = 1024 * V_{IN}$$

where: V_{IN} = analog input voltage
 V_{DD} = supply voltage



5.0 SERIAL COMMUNICATIONS

5.1 Overview

Communication with the MCP3002 is done using a standard SPI-compatible serial interface. Initiating communication with the device is done by bringing the CS line low. See Figure 5-1. If the device was powered up with the CS pin low, it must be brought high and back low to initiate communication. The first clock received with CS low and DIN high will constitute a start bit. The SGL/DIFF bit and the ODD/SIGN bit follow the start bit and are used to select the input channel configuration. The SGL/DIFF is used to select single ended or pseudo-differential mode. The ODD/SIGN bit selects which channel is used in single ended mode, and is used to determine polarity in psuedo-differential mode. Following the ODD/SIGN bit, the MSBF bit is transmitted to and is used to enable the LSB first format for the device. If the MSBF bit is low, then the data will come from the device in MSB first format and any further clocks with CS low, will cause the device to output zeros. If the MSBF bit is high, then the device will output the converted word LSB first *after* the word has been transmitted in the MSB first format. Table 5-1 shows the configuration bits for the MCP3002. The device will begin to sample the analog input on the second rising edge of the clock, after the start bit has been received. The sample period will end on the falling edge of the third clock following the start bit.

On the falling edge of the clock for the MSBF bit, the device will output a low null bit. The next sequential 10 clocks will output the result of the conversion with MSB first as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 10 data bits have been transmitted and the device continues to receive clocks while the CS is held low (and the MSBF bit is high), the device will output the conversion result LSB first as shown in Figure 5-2. If more clocks are provided to the device while CS is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring CS low and clock in leading zeros on the D_{IN} line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.1 for more details on using the MCP3002 devices with hardware SPI ports.

If it is desired, the CS can be raised to end the conversion period at any time during the transmission. Faster conversion rates can be obtained by using this technique if not all the bits are captured before starting a

new cycle. Some system designers use this method by capturing only the highest order 8 bits and ‘throwing away’ the lower 2 bits.

	CONFIG BITS		CHANNEL SELECTION		GND
	SGL/ DIFF	ODD/ SIGN	0	1	
SINGLE ENDED MODE	1	0	+		-
	1	1		+	-
PSEUDO-DIFFERENTIAL MODE	0	0	IN+	IN-	
	0	1	IN-	IN+	

TABLE 5-1: Configuration Bits for the MCP3002.

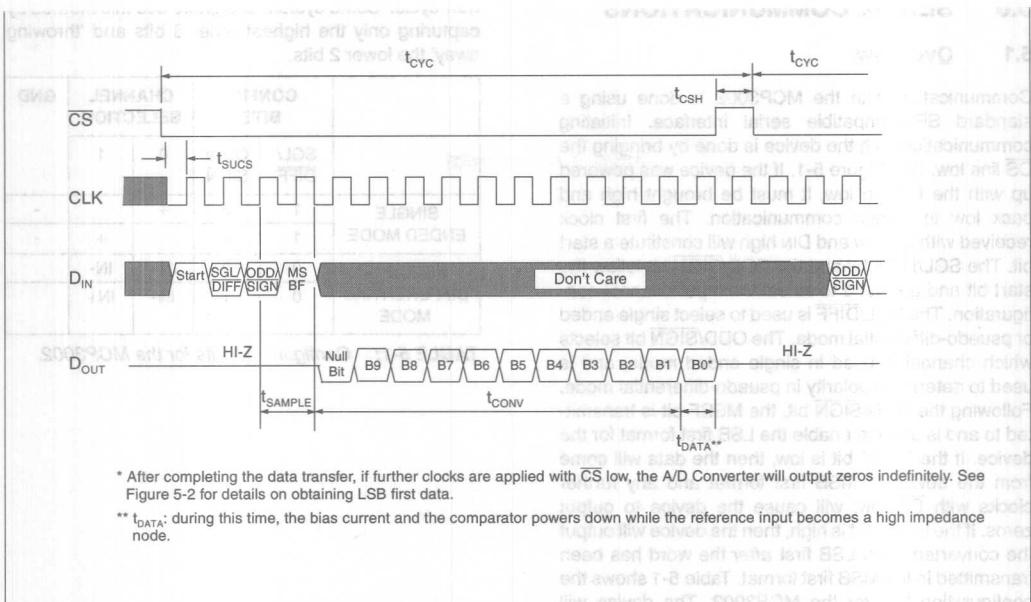


FIGURE 5-1: Communication with the MCP3002 using MSB first format only.

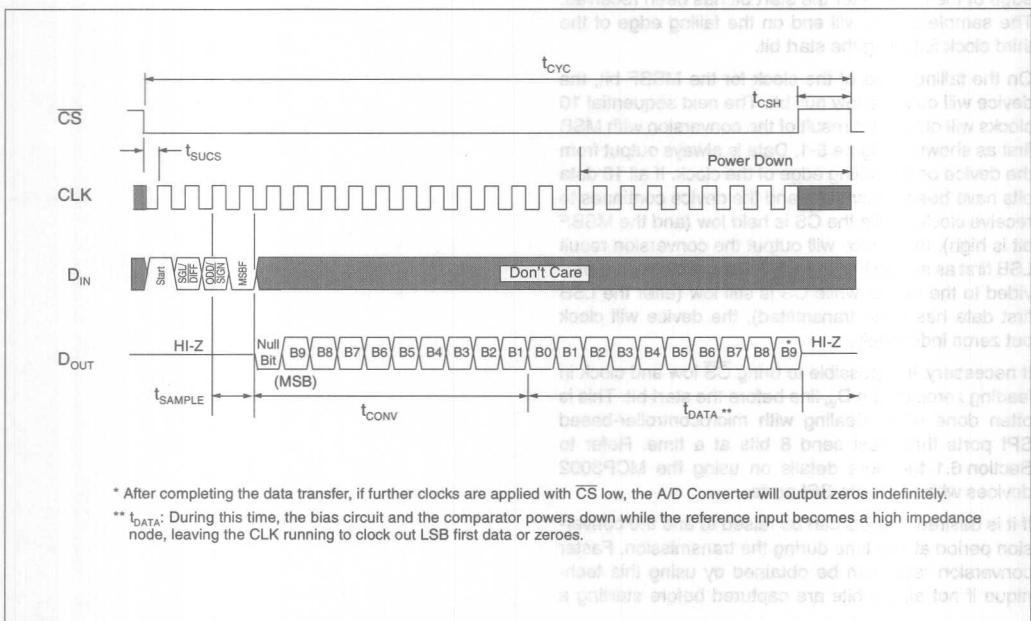


FIGURE 5-2: Communication with MCP3002 using LSB first format.

6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3002 with Microcontroller (MCU) SPI Ports

With most microcontroller SPI ports, it is required to send groups of eight bits. It is also required that the microcontroller SPI port be configured to clock out data on the falling edge of clock and latch data in on the rising edge. Depending on how communication routines are used, it is very possible that the number of clocks required for communication will not be a multiple of eight. Therefore, it may be necessary for the MCU to send more clocks than are actually required. This is usually done by sending 'leading zeros' before the start bit, which are ignored by the device. As an example, Figure 6-1 and Figure 6-2 show how the MCP3002 can

be interfaced to a MCU with a hardware SPI port. Figure 6-1 depicts the operation shown in SPI Mode 0,0, which requires that the SCLK from the MCU idles in the 'low' state, while Figure 6-2 shows the similar case of SPI Mode 1,1 where the clock idles in the 'high' state.

As shown in Figure 6-1, the first byte transmitted to the A/D Converter contains one leading zero before the start bit. Arranging the leading zero this way produces the output 10 bits to fall in positions easily manipulated by the MCU. When the first 8 bits are transmitted to the device, the MSB data bit is clocked out of the A/D Converter on the falling edge of clock number 6. After the second eight clocks have been sent to the device, the receive register will contain the lowest order eight bits of the conversion results. Easier manipulation of the converted data can be obtained by using this method.

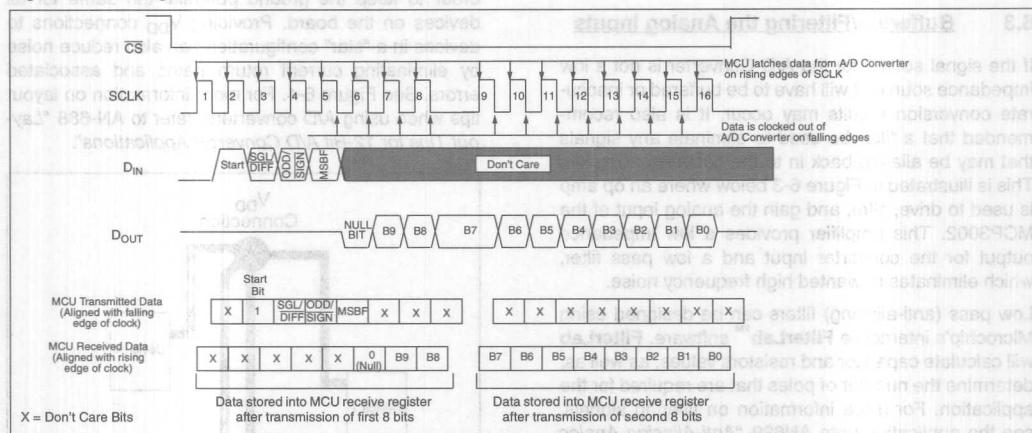


FIGURE 6-1: SPI Communication with the MCP3002 using 8-bit segments (Mode 0,0: SCLK idles low).

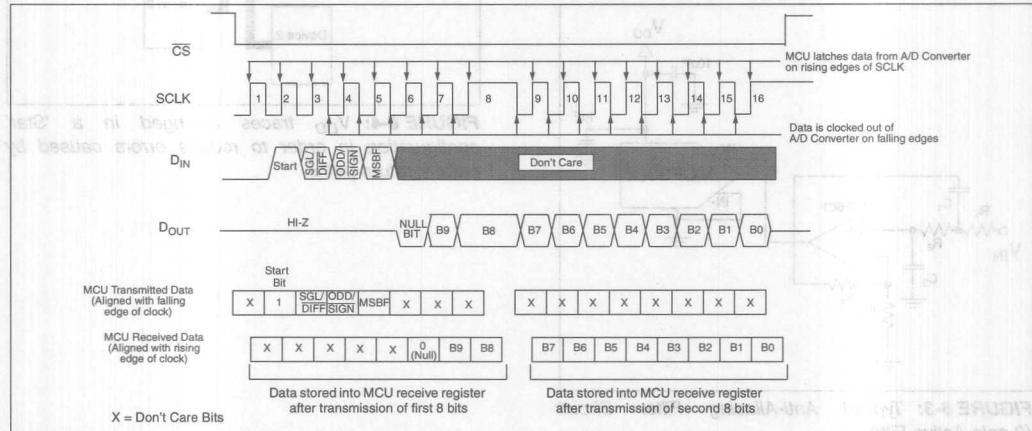


FIGURE 6-2: SPI Communication with the MCP3002 using 8-bit segments (Mode 1,1: SCLK idles high).

MCP3002

6.2 Maintaining Minimum Clock Speed

When the MCP3002 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample cap while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample cap for 700 μ s at V_{DD} = 2.7V and 1.5ms at V_{DD} = 5V. This means that at V_{DD} = 2.7V, the time it takes to transmit the 1.5 clocks for the sample period and the 10 clocks for the actual conversion must not exceed 700 μ s. Failure to meet this criteria may induce linearity errors into the conversion outside the rated specifications.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the A/D Converter is not a low impedance source, it will have to be buffered or inaccurate conversion results may occur. It is also recommended that a filter be used to eliminate any signals that may be aliased back in to the conversion results. This is illustrated in Figure 6-3 below where an op amp is used to drive, filter, and gain the analog input of the MCP3002. This amplifier provides a low impedance output for the converter input and a low pass filter, which eliminates unwanted high frequency noise.

Low pass (anti-aliasing) filters can be designed using Microchip's interactive **FilterLab™** software. **FilterLab** will calculate capacitor and resistors values, as well as, determine the number of poles that are required for the application. For more information on filtering signals, see the application note AN699 "Anti-Aliasing Analog Filters for Data Acquisition Systems".

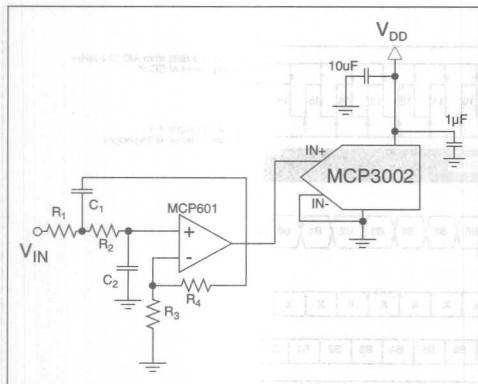


FIGURE 6-3: Typical Anti-Aliasing Filter Circuit (2 pole Active Filter).

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of 1 μ F is recommended.

Digital and analog traces should be separated as much as possible on the board and no traces should run underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating current return paths and associated errors. See Figure 6-4. For more information on layout tips when using A/D converters, refer to AN-688 "Layout Tips for 12-Bit A/D Converter Applications".

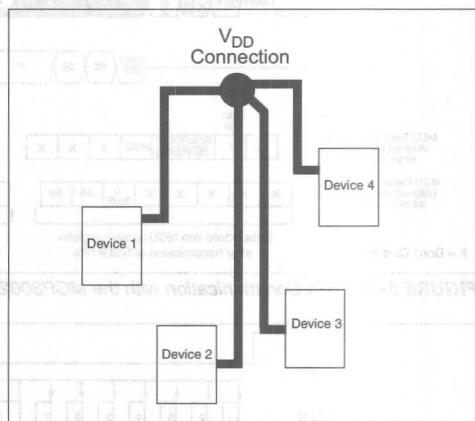


FIGURE 6-4: V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.

MCP3002 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

MCP3002 - X /X	
	Package: P = PDIP (8 lead) SN = SOIC (150 mil Body), 8 lead ST = TSSOP, 8 lead
	Temperature Range: I = -40°C to +85°C
	Device: MCP3002 = 10-Bit Serial A/D Converter MCP3002T = 10-Bit Serial A/D Converter on tape and reel (SOIC and TSSOP packages only)

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.



MICROCHIP

MCP3004/3008

2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI™ Serial Interface

FEATURES

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0, 0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200ksps max. sampling rate at $V_{DD} = 5V$
- 75ksps max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
 - 5nA typical standby current, 2 μ A max.
 - 500 μ A max. active current at 5V
- Industrial temp range: -40°C to +85°C
- Available in PDIP, SOIC and TSSOP packages

APPLICATIONS

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

DESCRIPTION

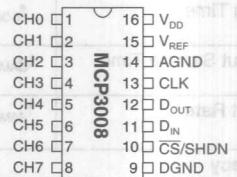
The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analog-to-Digital (A/D) Converters with on-board sample and hold circuitry. The MCP3004 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are specified at ± 1 LSB. Communication with the devices is done using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 200ksps. The MCP3004/3008 devices operate over a broad voltage range (2.7V - 5.5V). Low current design permits operation with typical standby currents of only 5nA and typical active currents of 320 μ A. The MCP3004 is offered in 14-pin PDIP, 150mil SOIC and TSSOP packages, and the MCP3008 is offered in 16-pin PDIP and SOIC packages.

PACKAGE TYPES

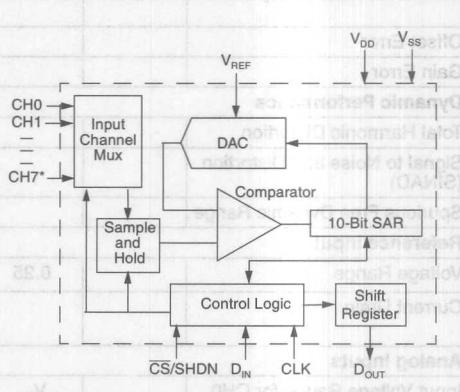
PDIP, SOIC, TSSOP



PDIP, SOIC



FUNCTIONAL BLOCK DIAGRAM



*Note: Channels 4-7 available on MCP3008 Only

3

Datasheets

MCP3004/3008

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to $V_{DD} + 0.6V$
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	> 4kV

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
V_{DD}	+2.7V to 5.5V Power Supply
DGND	Digital Ground
AGND	Analog Ground
CH0-CH7	Analog Inputs
CLK	Serial Clock
D_{IN}	Serial Data In
D_{OUT}	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input
V_{REF}	Reference Voltage Input

ELECTRICAL CHARACTERISTICS

All parameters apply at $V_{DD} = 5V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 200\text{ksps}$ and $f_{CLK} = 18 \times f_{SAMPLE}$, unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_{AMB} = 25^{\circ}C$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Conversion Rate						
Conversion Time	t_{CONV}			10	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}			200 75	ksps ksps	$V_{DD} = V_{REF} = 5V$ $V_{DD} = V_{REF} = 2.7V$
DC Accuracy						
Resolution			10		bits	
Integral Nonlinearity	INL		± 0.5	± 1	LSB	
Differential Nonlinearity	DNL		± 0.25	± 1	LSB	No missing codes over temperature
Offset Error				± 1.5	LSB	
Gain Error				± 1	LSB	
Dynamic Performance						
Total Harmonic Distortion			-76		dB	$VIN = 0.1V$ to $4.9V$ @ $1kHz$
Signal to Noise and Distortion (SINAD)			61		dB	$VIN = 0.1V$ to $4.9V$ @ $1kHz$
Spurious Free Dynamic Range			78		dB	$VIN = 0.1V$ to $4.9V$ @ $1kHz$
Reference Input						
Voltage Range		0.25		V_{DD}	V	Note 2
Current Drain			100 0.001	150 3	μA	$CS = V_{DD} = 5V$
Analog Inputs						
Input Voltage Range for CH0 or CH1 in Single-Ended Mode		V_{SS}		V_{REF}	V	
Input Voltage Range for IN+ In pseudo-differential Mode		IN-		$V_{REF} + IN$		
Input Voltage Range for IN- In pseudo-differential Mode		$V_{SS} - 100$		$V_{SS} + 100$	mV	
Leakage Current			0.001	± 1	μA	
Switch Resistance			1K		Ω	See Figure 4-1
Sample Capacitor			20		pF	See Figure 4-1

ELECTRICAL CHARACTERISTICS (CONTINUED)

All parameters apply at $V_{DD} = 5V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 200\text{ksps}$ and $f_{CLK} = 18 \times f_{SAMPLE}$, unless otherwise noted. Typical values apply for $V_{DD} = 5V$, $T_{AMB} = 25^{\circ}C$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Digital Input/Output						
Data Coding Format			Straight Binary			
High Level Input Voltage	V_{IH}	0.7 V_{DD}			V	
Low Level Input Voltage	V_{IL}			0.3 V_{DD}	V	
High Level Output Voltage	V_{OH}	4.1			V	$I_{OH} = -1\text{mA}$, $V_{DD} = 4.5\text{V}$
Low Level Output Voltage	V_{OL}			0.4	V	$I_{OL} = 1\text{mA}$, $V_{DD} = 4.5\text{V}$
Input Leakage Current	I_{LI}	-10		10	μA	$V_{IN} = V_{SS}$ or V_{DD}
Output Leakage Current	I_{LO}	-10		10	μA	$V_{OUT} = V_{SS}$ or V_{DD}
Pin Capacitance (All Inputs/Outputs)	C_{IN} , C_{OUT}			10	pF	$V_{DD} = 5.0\text{V}$ (Note 1) $T_{AMB} = 25^{\circ}\text{C}$, $f = 1\text{ MHz}$
Timing Parameters						
Clock Frequency	f_{CLK}			3.6 1.35	MHz MHz	$V_{DD} = 5\text{V}$ (Note 3) $V_{DD} = 2.7\text{V}$ (Note 3)
Clock High Time	t_{H1}	125			ns	
Clock Low Time	t_{L1}	125			ns	
CS Fall To First Rising CLK Edge	t_{SUCS}	100			ns	
CS Fall To Falling CLK Edge	t_{CSD}			0	ns	
Data Input Setup Time	t_{SU}			50	ns	
Data Input Hold Time	t_{HD}			50	ns	
CLK Fall To Output Data Valid	t_{DO}			125 200	ns ns	$V_{DD} = 5\text{V}$, See Figure 1-2 $V_{DD} = 2.7$, See Figure 1-2
CLK Fall To Output Enable	t_{EN}			125 200	ns ns	$V_{DD} = 5\text{V}$, See Figure 1-2 $V_{DD} = 2.7$, See Figure 1-2
CS Rise To Output Disable	t_{DIS}			100	ns	See Test Circuits, Figure 1-2
CS Disable Time	t_{CSH}	270			ns	
D_{OUT} Rise Time	t_R			100	ns	See Test Circuits, Figure 1-2 (Note 1)
D_{OUT} Fall Time	t_F			100	ns	See Test Circuits, Figure 1-2 (Note 1)
Power Requirements						
Operating Voltage	V_{DD}	2.7		5.5	V	
Operating Current	I_{DD}		425 225	550	μA	$V_{DD}=V_{REF}=5\text{V}$, D_{OUT} unloaded $V_{DD}=V_{REF}=2.7\text{V}$, D_{OUT} unloaded
Standby Current	I_{DDS}		0.005	2	μA	$CS = V_{DD} = 5.0\text{V}$

Note 1: This parameter is guaranteed by characterization and not 100% tested.

Note 2: See graphs that relate linearity performance to V_{REF} levels.

Note 3: Because the sample cap will eventually lose charge, effective clock rates below 10kHz can affect linearity performance, especially at elevated temperatures. See Section 6.2 for more information.

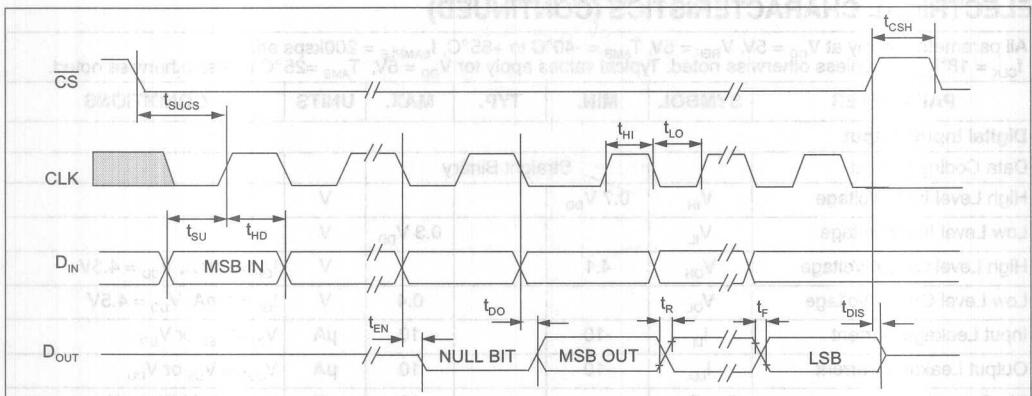


FIGURE 1-1: Serial Interface Timing.

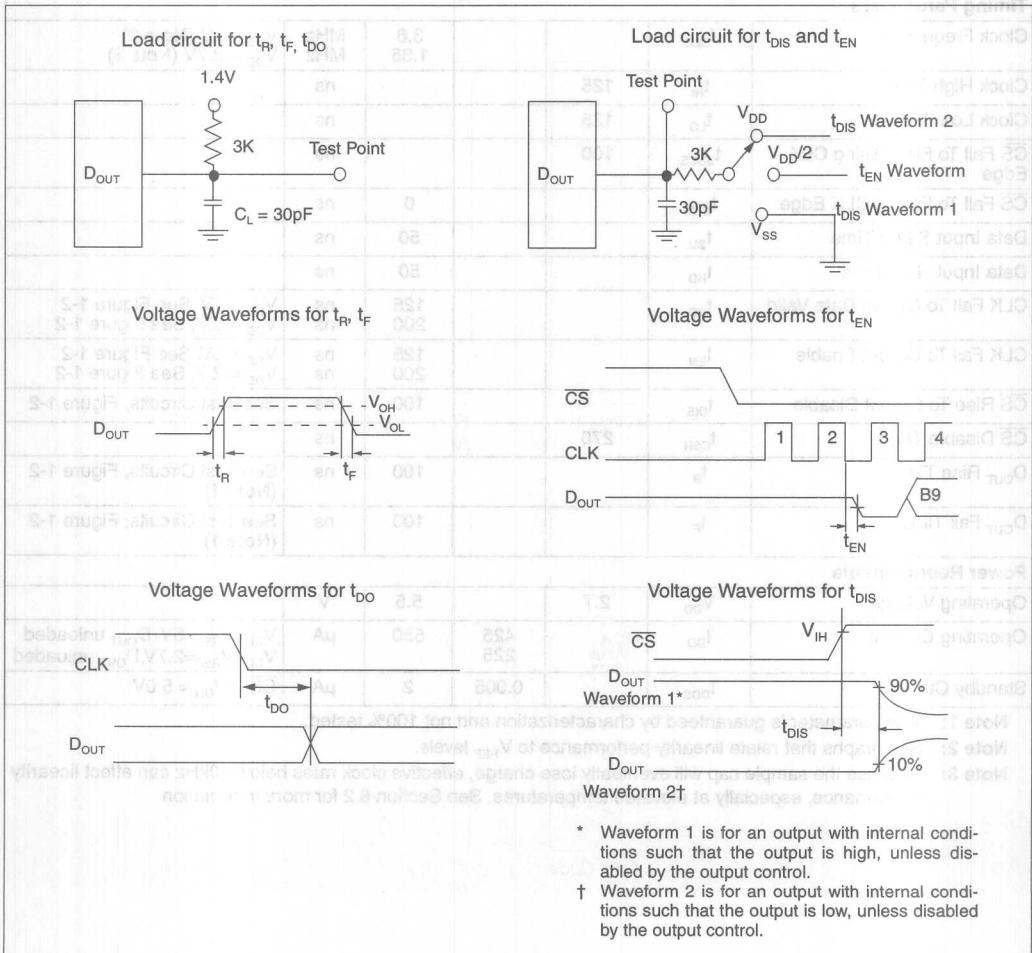


FIGURE 1-2: Test Circuits.

2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18 * f_{SAMPLE}$, $T_A = 25^\circ C$

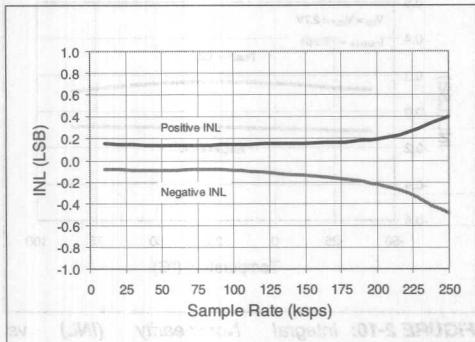


FIGURE 2-1: Integral Nonlinearity (INL) vs. Sample Rate.

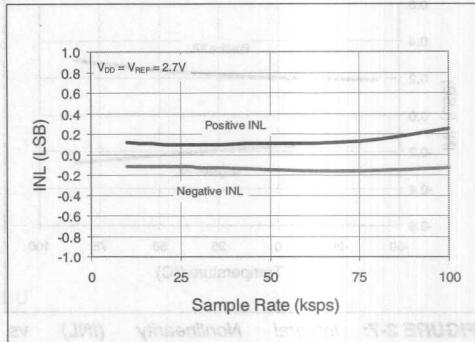


FIGURE 2-4: Integral Nonlinearity (INL) vs. Sample Rate ($V_{DD} = 2.7V$).

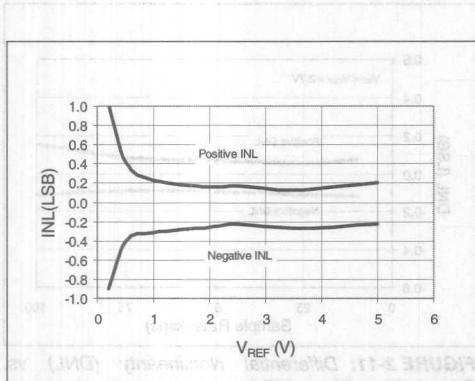


FIGURE 2-2: Integral Nonlinearity (INL) vs. V_{REF} .

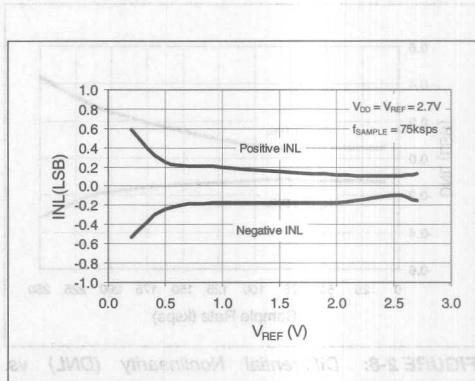


FIGURE 2-5: Integral Nonlinearity (INL) vs. V_{REF} ($V_{DD} = 2.7V$).

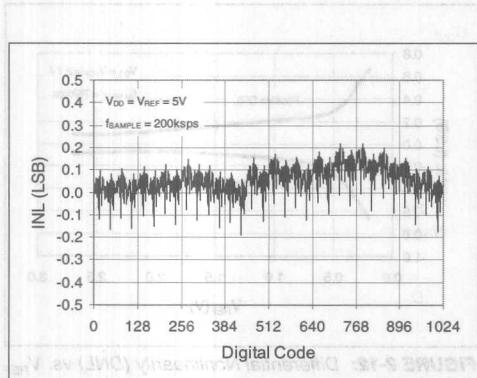


FIGURE 2-3: Integral Nonlinearity (INL) vs. Code (Representative Part).

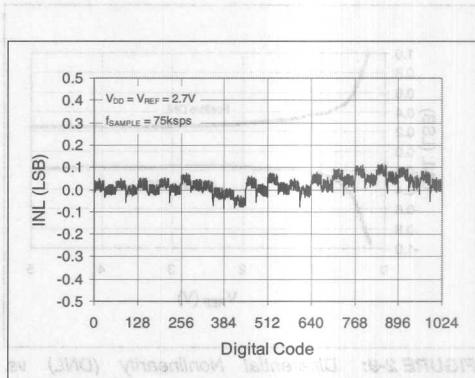


FIGURE 2-6: Integral Nonlinearity (INL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

MCP3004/3008

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = 25^\circ C$

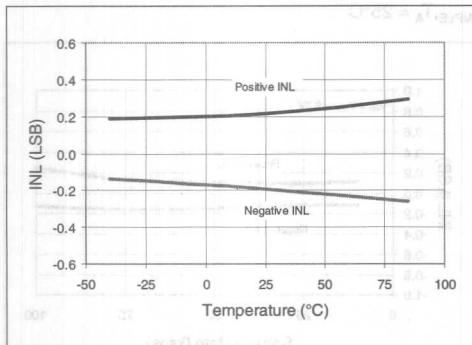


FIGURE 2-7: Integral Nonlinearity (INL) vs. Temperature.

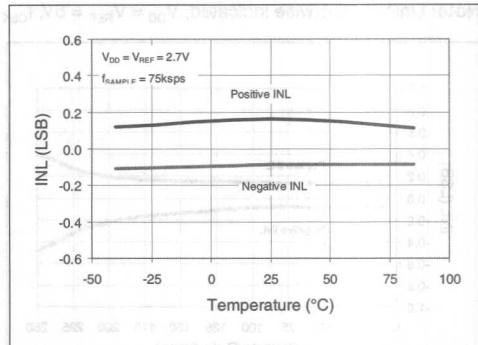


FIGURE 2-10: Integral Nonlinearity (INL) vs. Temperature ($V_{DD} = 2.7V$).

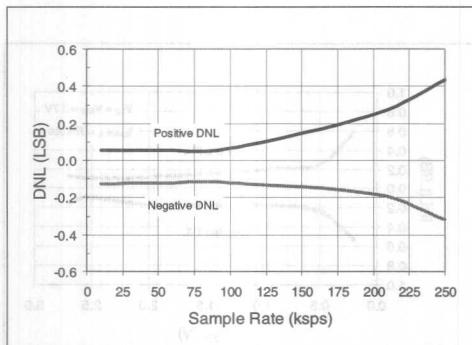


FIGURE 2-8: Differential Nonlinearity (DNL) vs. Sample Rate.

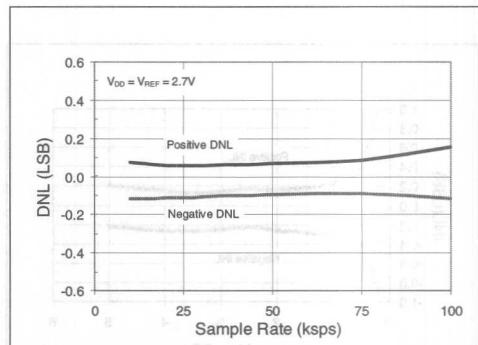


FIGURE 2-11: Differential Nonlinearity (DNL) vs. Sample Rate ($V_{DD} = 2.7V$).

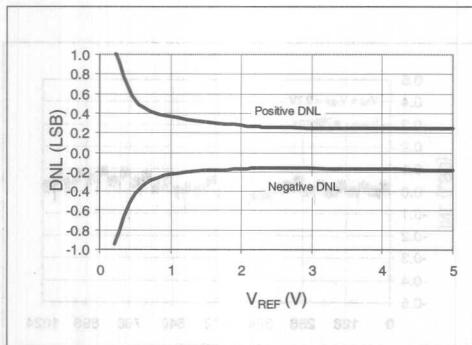


FIGURE 2-9: Differential Nonlinearity (DNL) vs. V_{REF} .

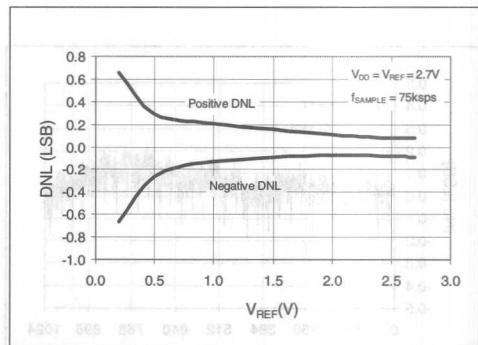


FIGURE 2-12: Differential Nonlinearity (DNL) vs. V_{REF} ($V_{DD} = 2.7V$).

MCP3004/3008

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = 25^\circ C$

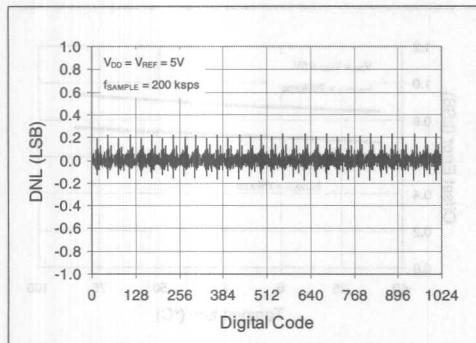


FIGURE 2-13: Differential Nonlinearity (DNL) vs. Code (Representative Part).

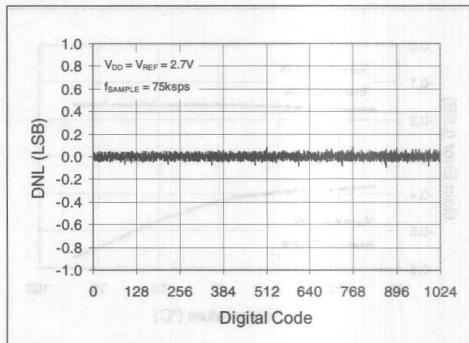


FIGURE 2-16: Differential Nonlinearity (DNL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

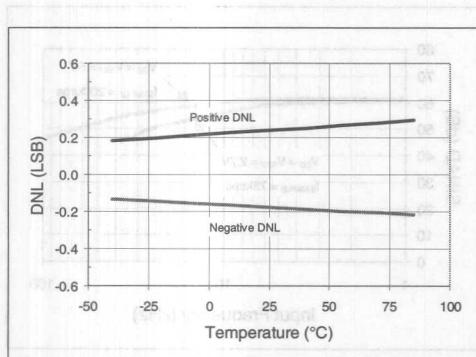


FIGURE 2-14: Differential Nonlinearity (DNL) vs. Temperature.

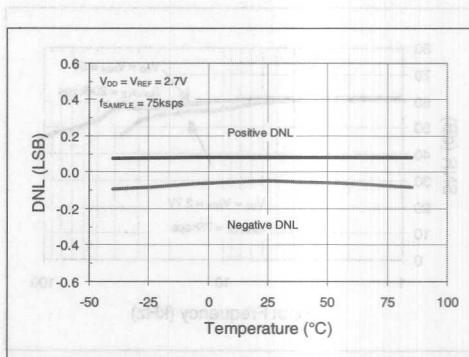


FIGURE 2-17: Differential Nonlinearity (DNL) vs. Temperature ($V_{DD} = 2.7V$).

3

Datasheets

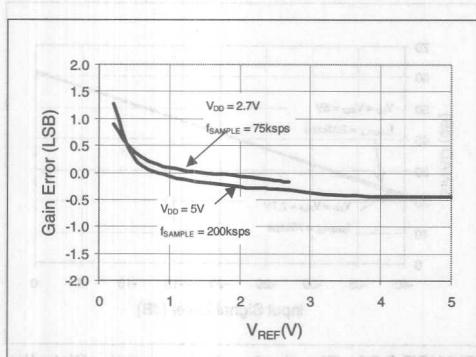


FIGURE 2-15: Gain Error vs. V_{REF} .

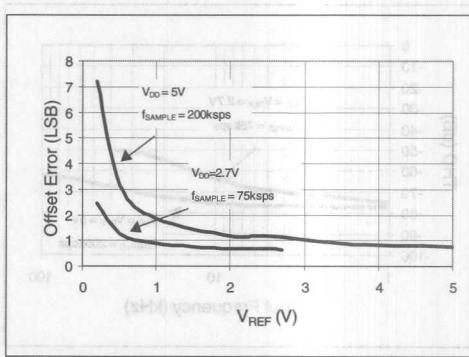


FIGURE 2-18: Offset Error vs. V_{REF} .

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18 * f_{SAMPLE}$, $T_A = 25^\circ C$, $\Delta V_{DD} = \Delta V_{REF} = \pm 5\%$, $f_{RAMP} = 100\text{Hz}$, $f_{SIN} = 10\text{kHz}$, $f_{NOISE} = 100\text{Hz}$, $f_{THD} = 10\text{kHz}$

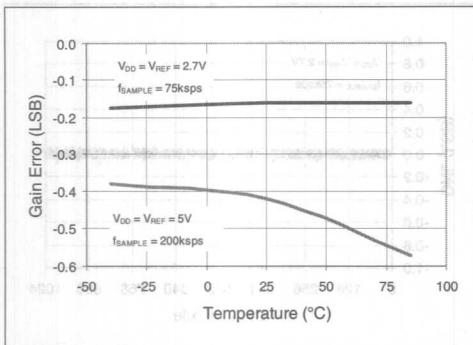


FIGURE 2-19: Gain Error vs. Temperature.

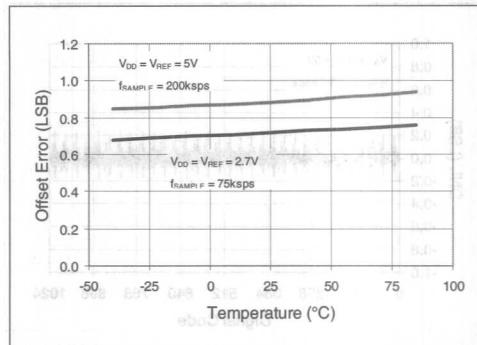


FIGURE 2-22: Offset Error vs. Temperature.

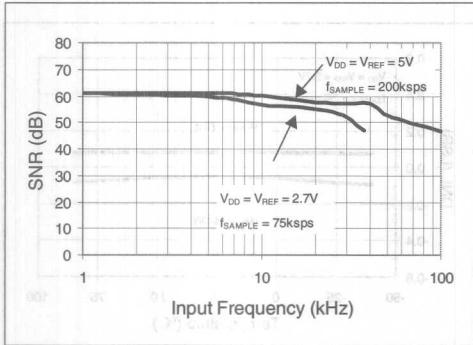


FIGURE 2-20: Signal to Noise (SNR) vs. Input Frequency.

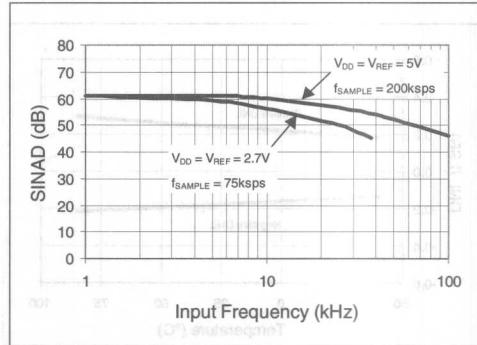


FIGURE 2-23: Signal to Noise and Distortion (SINAD) vs. Input Frequency.

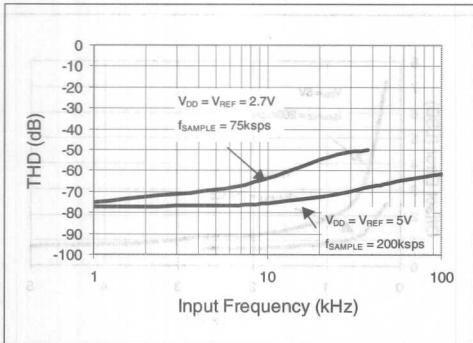


FIGURE 2-21: Total Harmonic Distortion (THD) vs. Input Frequency.

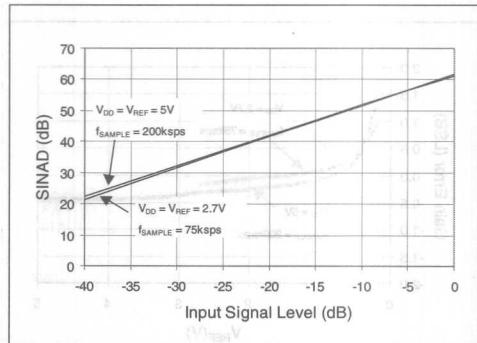


FIGURE 2-24: Signal to Noise and Distortion (SINAD) vs. Input Signal Level.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = 25^\circ C$

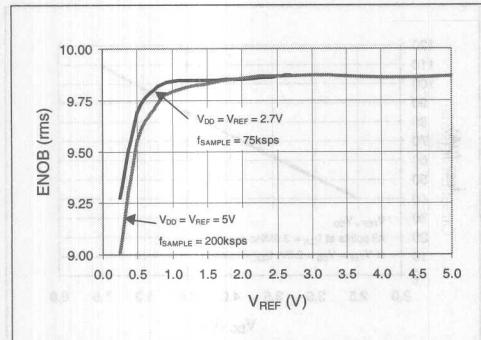


FIGURE 2-25: Effective Number of Bits (ENOB) vs. V_{REF} .

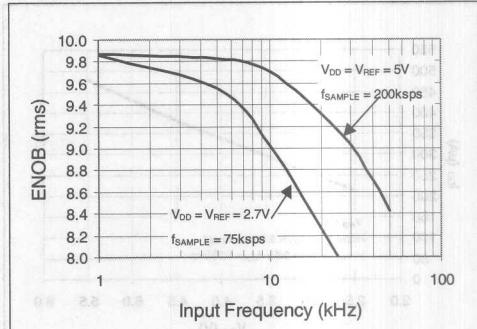


FIGURE 2-28: Effective Number of Bits (ENOB) vs. Input Frequency.

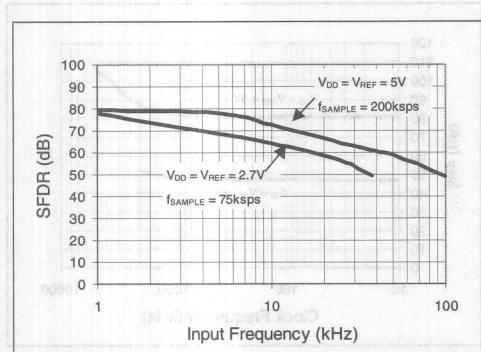


FIGURE 2-26: Spurious Free Dynamic Range (SFDR) vs. Input Frequency.

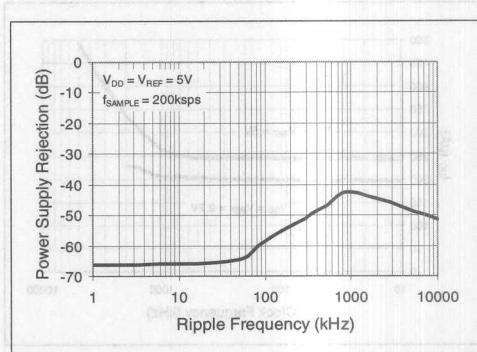


FIGURE 2-29: Power Supply Rejection (PSR) vs. Ripple Frequency.

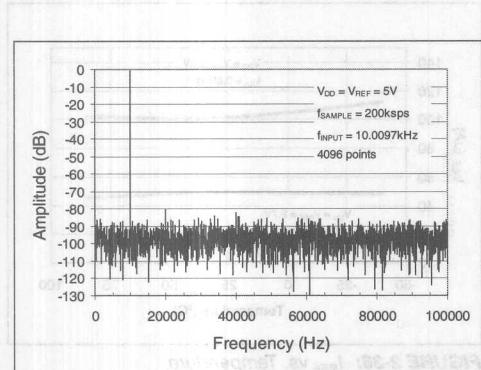


FIGURE 2-27: Frequency Spectrum of 10kHz input (Representative Part).

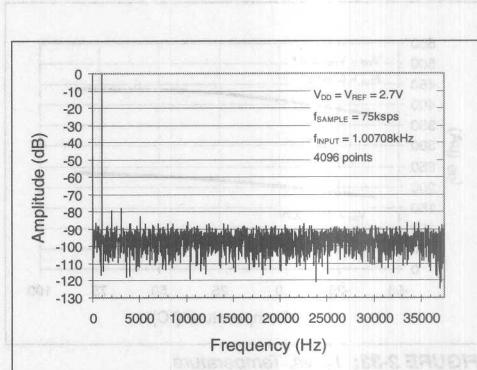


FIGURE 2-30: Frequency Spectrum of 1kHz input (Representative Part, $V_{DD} = 2.7V$).

MCP3004/3008

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18 * f_{SAMPLE}$, $T_A = 25^\circ C$

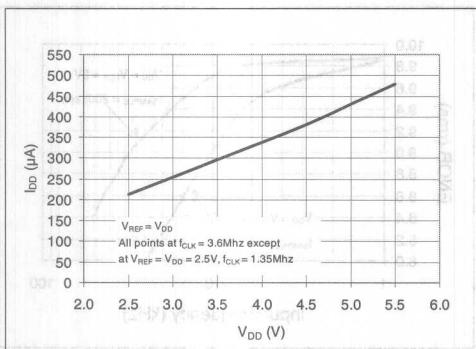


FIGURE 2-31: I_{DD} vs. V_{DD} .

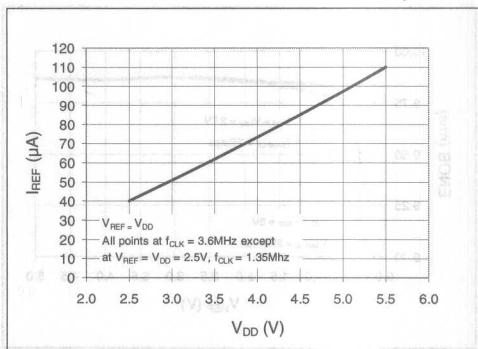


FIGURE 2-34: I_{REF} vs. V_{DD} .

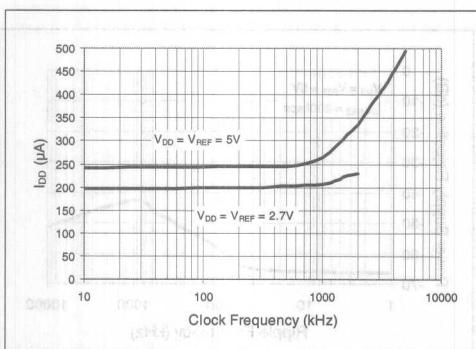


FIGURE 2-32: I_{DD} vs. Clock Frequency.

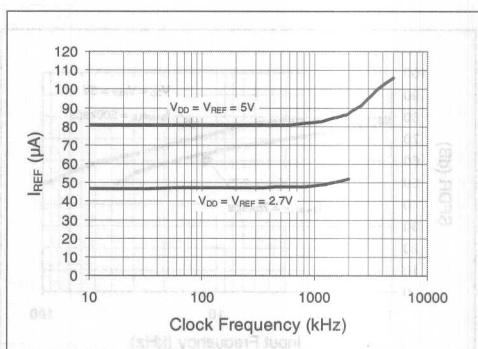


FIGURE 2-35: I_{REF} vs. Clock Frequency.

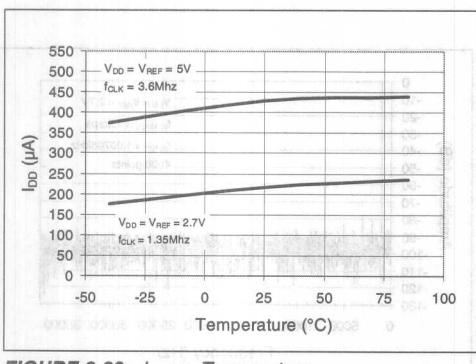


FIGURE 2-33: I_{DD} vs. Temperature.

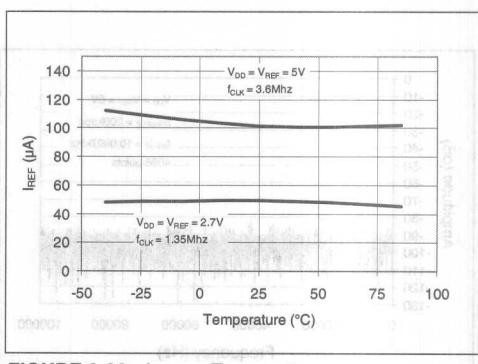


FIGURE 2-36: I_{REF} vs. Temperature.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18 * f_{SAMPLE}$, $T_A = 25^\circ C$

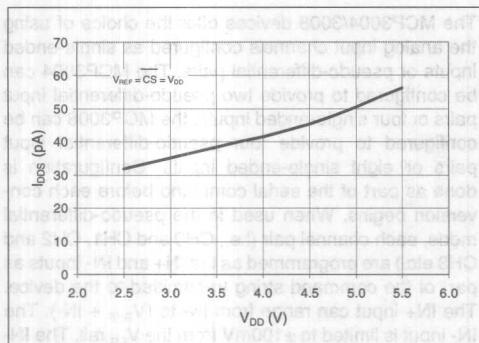


FIGURE 2-37: I_{DDS} vs. V_{DD} .

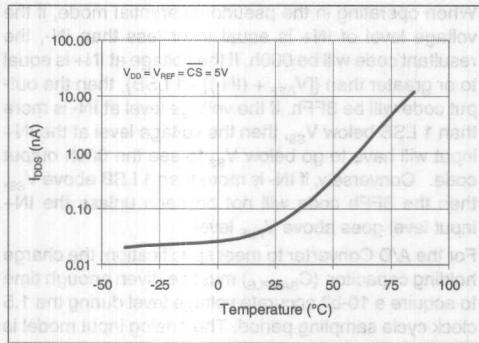


FIGURE 2-38: I_{DDS} vs. Temperature.

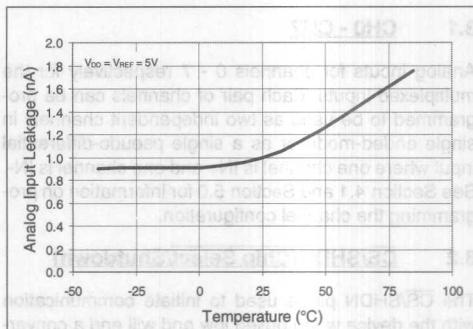


FIGURE 2-39: Analog Input Leakage Current vs. Temperature.

3.0 PIN DESCRIPTIONS

3.1 CH0 - CH7

Analog inputs for channels 0 - 7 respectively for the multiplexed inputs. Each pair of channels can be programmed to be used as two independent channels in single ended-mode or as a single pseudo-differential input where one channel is IN+ and one channel is IN-. See Section 4.1 and Section 5.0 for information on programming the channel configuration.

3.2 CS/SHDN(Chip Select/Shutdown)

The CS/SHDN pin is used to initiate communication pulled high. The CS/SHDN pin must be pulled high between conversions.

3.3 CLK (Serial Clock)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 6.2 for constraints on clock speed.

3.4 DIN (Serial Data Input)

The SPI port serial data input pin is used to load channel configuration data into the device.

3.5 DOUT (Serial Data output)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

3.6 AGND

Analog ground connection to internal analog circuitry.

3.7 DGND

Digital ground connection to internal digital circuitry.

4.0 DEVICE OPERATION

The MCP3004/3008 A/D Converters employ a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the first rising edge of the serial clock after CS has been pulled low. Following this sample time, the device uses the collected charge on the internal sample and hold capacitor to produce a serial 10-bit digital output code. Conversion rates of 100ksps are possible on the MCP3004/3008. See Section 6.2 for information on minimum clock rates. Communication with the device is done using a 4-wire SPI-compatible interface.

The MCP3004/3008 devices offer the choice of using the analog input channels configured as single-ended inputs or pseudo-differential pairs. The MCP3004 can be configured to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 can be configured to provide four pseudo-differential input pairs or eight single-ended inputs. Configuration is done as part of the serial command before each conversion begins. When used in the pseudo-differential mode, each channel pair (i.e., CH0 and CH1, CH2 and CH3 etc.) are programmed as the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ can't be used to cancel the IN- signal. The mon-mode noise which is present on both the IN+ and IN- inputs.

When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be 00h. If the voltage at IN+ is equal to or greater than {V_{REF} + (IN-) - 1 LSB}, then the output code will be 3FFh. If the voltage level at IN- is more than 1 LSB below V_{SS}, then the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS}, then the 3FFh code will not be seen unless the IN+ input level goes above V_{REF} level.

For the A/D Converter to meet specification, the charge holding capacitor, (C_{SAMPLE}) must be given enough time to acquire a 10-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

In this diagram it is shown that the source impedance (R_S) adds to the internal sampling switch (R_{ss}) impedance, directly affecting the time that is required to charge the capacitor, C_{SAMPLE}. Consequently, larger source impedances increase the offset, gain, and integral linearity errors of the conversion. See Figure 4-2.

4.2 Reference Input

For each device in the family, the reference input (V_{REF}) determines the analog input voltage range. As the reference input is reduced, the LSB size is reduced accordingly.

$$LSB\ Size = \frac{V_{REF}}{1024}$$

The theoretical digital output code produced by the A/D Converter is a function of the analog input signal and the reference input as shown below.

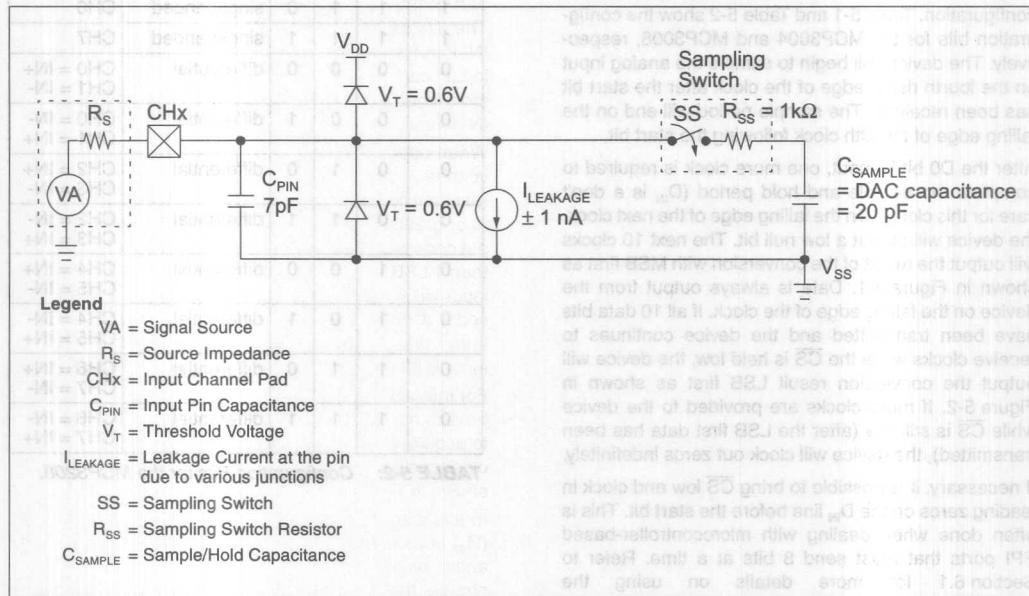


FIGURE 4-1: Analog Input Model

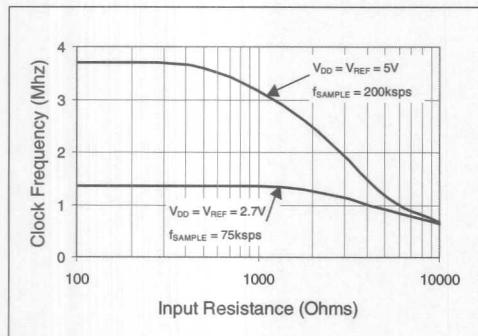


FIGURE 4-2: Maximum Clock Frequency vs. Input resistance (R_S) to maintain less than a 0.1LSB deviation in INL from nominal conditions.

$$Digital\ Output\ Code = \frac{1024 * V_{IN}}{V_{REF}}$$

where:

V_{IN} = analog input voltage

V_{REF} = reference voltage

When using an external voltage reference device, the system designer should always refer to the manufacturer's recommendations for circuit layout. Any instability in the operation of the reference device will have a direct effect on the operation of the A/D Converter.

Legend:

VA = Signal Source

R_S = Source Impedance

CHx = Input Channel Pad

C_{PIN} = Input Pin Capacitance

V_T = Threshold Voltage

$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions

SS = Sampling Switch

R_{SS} = Sampling Switch Resistor

C_{SAMPLE} = Sample/Hold Capacitance

$= 20\text{ pF}$

Legend:

VA = Signal Source

R_S = Source Impedance

CHx = Input Channel Pad

C_{PIN} = Input Pin Capacitance

V_T = Threshold Voltage

$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions

SS = Sampling Switch

R_{SS} = Sampling Switch Resistor

C_{SAMPLE} = Sample/Hold Capacitance

$= 20\text{ pF}$

Legend:

VA = Signal Source

R_S = Source Impedance

CHx = Input Channel Pad

C_{PIN} = Input Pin Capacitance

V_T = Threshold Voltage

$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions

SS = Sampling Switch

R_{SS} = Sampling Switch Resistor

C_{SAMPLE} = Sample/Hold Capacitance

$= 20\text{ pF}$

Legend:

VA = Signal Source

R_S = Source Impedance

CHx = Input Channel Pad

C_{PIN} = Input Pin Capacitance

V_T = Threshold Voltage

$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions

SS = Sampling Switch

R_{SS} = Sampling Switch Resistor

C_{SAMPLE} = Sample/Hold Capacitance

$= 20\text{ pF}$

Legend:

VA = Signal Source

R_S = Source Impedance

CHx = Input Channel Pad

C_{PIN} = Input Pin Capacitance

V_T = Threshold Voltage

$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions

SS = Sampling Switch

R_{SS} = Sampling Switch Resistor

C_{SAMPLE} = Sample/Hold Capacitance

$= 20\text{ pF}$

SELECTED CHANNEL	INPUT CONFIGURATION	GND	
		CH0	CH1
CH0	single ended	0	X
CH1	single ended	1	X
CH2	single ended	0	X
CH3	single ended	1	X
CH0 = IN+	differential	0	X
CH1 = IN-		1	X
CH2 = IN+	differential	1	X
CH3 = IN-		0	X
CH0 = IN+	differential	0	X
CH1 = IN-		1	X
CH2 = IN+	differential	1	X
CH3 = IN-		0	X

MCP3004/3008

5.0 SERIAL COMMUNICATIONS

Communication with the MCP3004/3008 devices is done using a standard SPI-compatible serial interface. Initiating communication with either device is done by bringing the \overline{CS} line low. See Figure 5-1. If the device was powered up with the \overline{CS} pin low, it must be brought high and back low to initiate communication. The first clock received with \overline{CS} low and D_{IN} high will constitute a start bit. The SGL/DIFF bit follows the start bit and will determine if the conversion will be done using single ended or differential input mode. The next three bits (D0, D1 and D2) are used to select the input channel configuration. Table 5-1 and Table 5-2 show the configuration bits for the MCP3004 and MCP3008, respectively. The device will begin to sample the analog input on the fourth rising edge of the clock after the start bit has been received. The sample period will end on the falling edge of the fifth clock following the start bit.

After the D0 bit is input, one more clock is required to complete the sample and hold period (D_{IN} is a don't care for this clock). On the falling edge of the next clock, the device will output a low null bit. The next 10 clocks will output the result of the conversion with MSB first as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 10 data bits have been transmitted and the device continues to receive clocks while the \overline{CS} is held low, the device will output the conversion result LSB first as shown in Figure 5-2. If more clocks are provided to the device while \overline{CS} is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring \overline{CS} low and clock in leading zeros on the D_{IN} line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.1 for more details on using the MCP3004/3008 devices with hardware SPI ports.

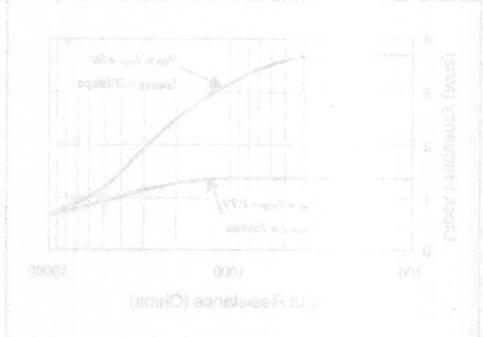
CONTROL BIT SELECTIONS				INPUT CONFIGURATION	CHANNEL SELECTION
SINGLE/ DIFF	D2*	D1	D0		
1	X	0	0	single ended	CH0
1	X	0	1	single ended	CH1
1	X	1	0	single ended	CH2
1	X	1	1	single ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	1	0	differential	CH4 = IN- CH5 = IN+
0	1	0	1	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

*D2 is don't care for MCP3004

TABLE 5-1: Configuration Bits for the MCP3204.

CONTROL BIT SELECTIONS				INPUT CONFIGURATION	CHANNEL SELECTION
SINGLE/ DIFF	D2	D1	D0		
1	0	0	0	single ended	CH0
1	0	0	1	single ended	CH1
1	0	1	0	single ended	CH2
1	0	1	1	single ended	CH3
1	1	0	0	single ended	CH4
1	1	0	1	single ended	CH5
1	1	1	0	single ended	CH6
1	1	1	1	single ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

TABLE 5-2: Configuration Bits for the MCP3208.



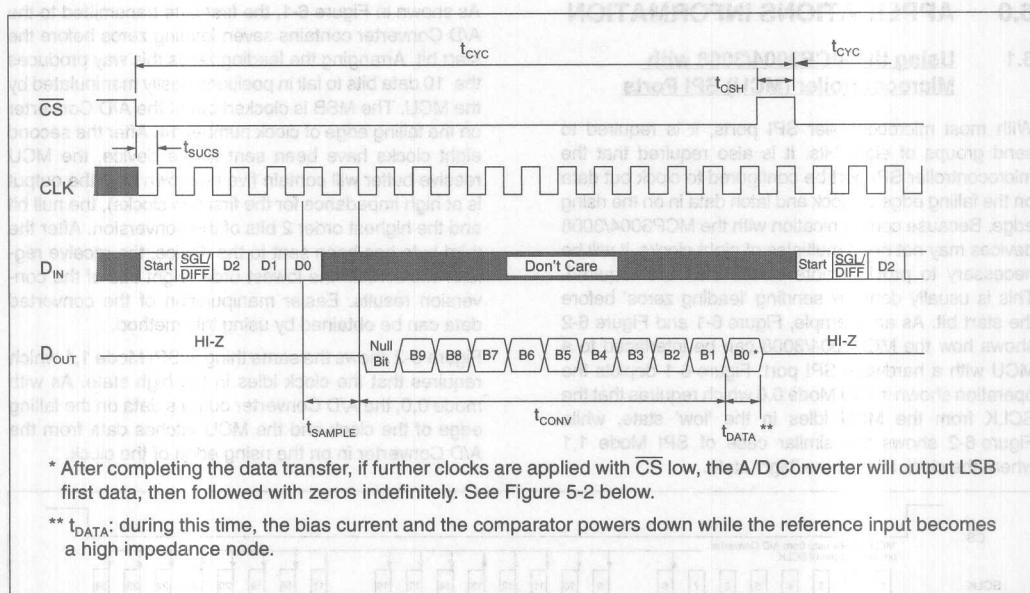


FIGURE 5-1: Communication with the MCP3204 or MCP3208.

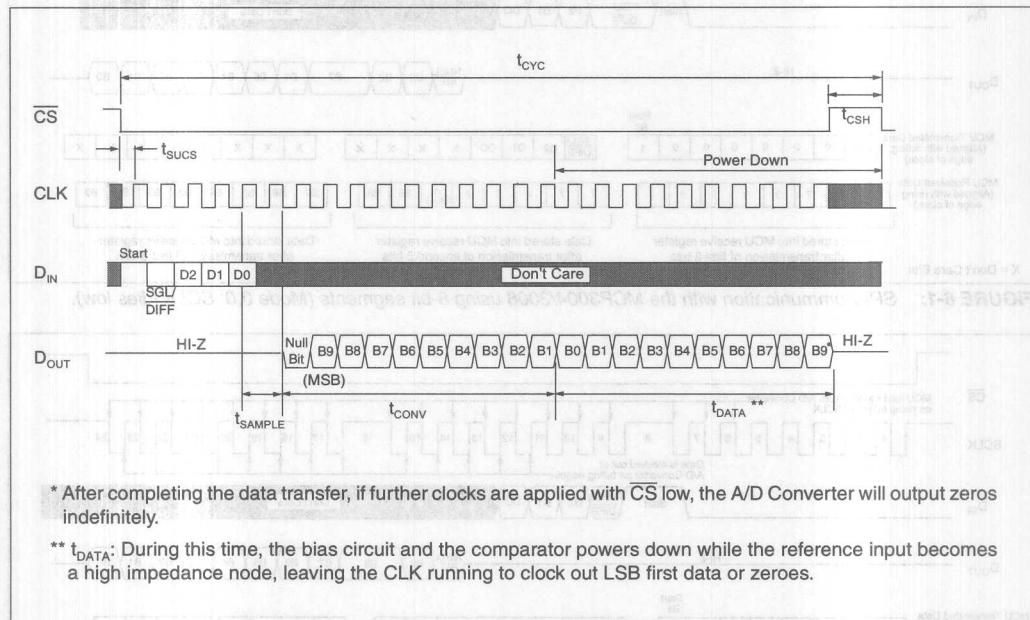
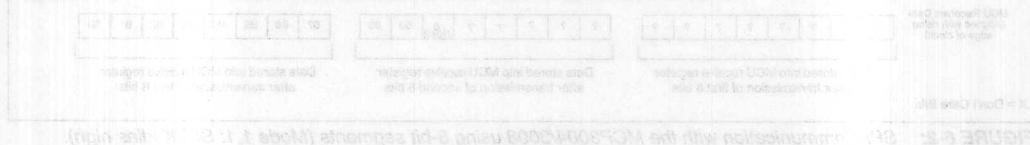


FIGURE 5-2: Communication with MCP3004 or MCP3008 in LSB First Format.



6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports

With most microcontroller SPI ports, it is required to send groups of eight bits. It is also required that the microcontroller SPI port be configured to clock out data on the falling edge of clock and latch data in on the rising edge. Because communication with the MCP3004/3008 devices may not need multiples of eight clocks, it will be necessary to provide more clocks than are required. This is usually done by sending 'leading zeros' before the start bit. As an example, Figure 6-1 and Figure 6-2 shows how the MCP3004/3008 can be interfaced to a MCU with a hardware SPI port. Figure 6-1 depicts the operation shown in SPI Mode 0,0 which requires that the SCLK from the MCU idles in the 'low' state, while Figure 6-2 shows the similar case of SPI Mode 1,1 where the clock idles in the 'high' state.

As shown in Figure 6-1, the first byte transmitted to the A/D Converter contains seven leading zeros before the start bit. Arranging the leading zeros this way produces the 10 data bits to fall in positions easily manipulated by the MCU. The MSB is clocked out of the A/D Converter on the falling edge of clock number 14. After the second eight clocks have been sent to the device, the MCU receive buffer will contain five unknown bits (the output is at high impedance for the first two clocks), the null bit and the highest order 2 bits of the conversion. After the third byte has been sent to the device, the receive register will contain the lowest order eight bits of the conversion results. Easier manipulation of the converted data can be obtained by using this method.

Figure 6-2 shows the same thing in SPI Mode 1,1 which requires that the clock idles in the high state. As with mode 0,0, the A/D Converter outputs data on the falling edge of the clock and the MCU latches data from the A/D Converter in on the rising edge of the clock.

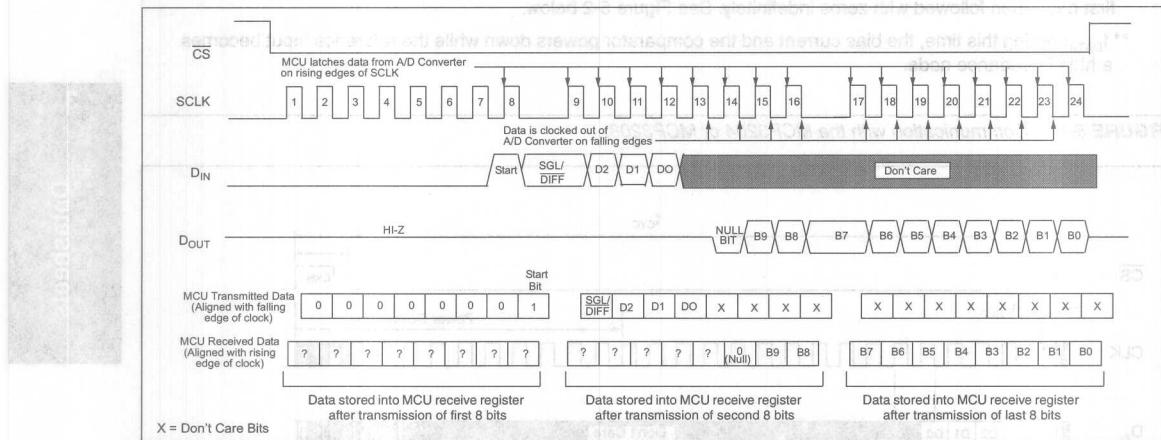


FIGURE 6-1: SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 0,0: SCLK idles low).

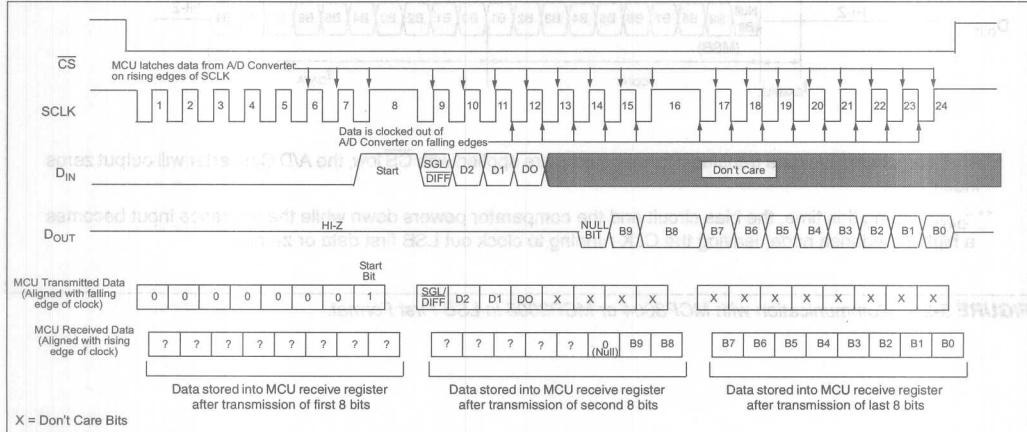


FIGURE 6-2: SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 1,1: SCLK idles high).

6.2 Maintaining Minimum Clock Speed

When the MCP3004/3008 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample capacitor while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample capacitor for at least 1.2ms after the sample period has ended. This means that the time between the end of the sample period and the time that all 10 data bits have been clocked out must not exceed 1.2ms (effective clock frequency of 10kHz). Failure to meet this criterion may induce linearity errors into the conversion outside the rated specifications. It should be noted that during the entire conversion cycle, the A/D Converter does not require a constant clock speed or duty cycle, as long as all timing specifications are met.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the A/D Converter is not a low impedance source, it will have to be buffered or inaccurate conversion results may occur. See Figure 4-2. It is also recommended that a filter be used to eliminate any signals that may be aliased back into the conversion results. This is illustrated in Figure 6-3 where an op amp is used to drive, filter and gain the analog input of the MCP3004/3008. This amplifier provides a low impedance source for the converter input and a low pass filter, which eliminates unwanted high frequency noise.

Low pass (anti-aliasing) filters can be designed using Microchip's free interactive **FilterLab™** software. **FilterLab** will calculate capacitor and resistors values, as well as determine the number of poles that are required for the application. For more information on filtering signals, see the application note AN699 "Anti-Aliasing Analog Filters for Data Acquisition Systems."

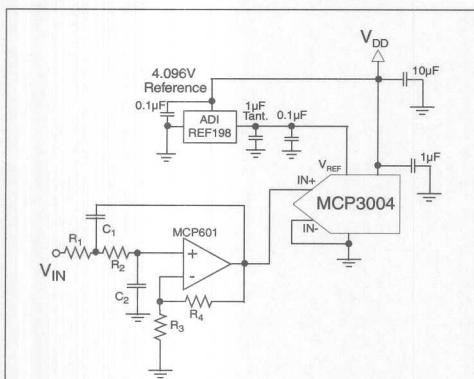


FIGURE 6-3: The MCP601 Operational Amplifier is used to implement a 2nd order anti-aliasing filter for the signal being converted by the MCP3004.

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of 1µF is recommended.

Digital and analog traces should be separated as much as possible on the board and no traces should run underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating return current paths and associated errors. See Figure 6-4. For more information on layout tips when using A/D Converters, refer to AN688 "Layout Tips for 12-Bit A/D Converter Applications".

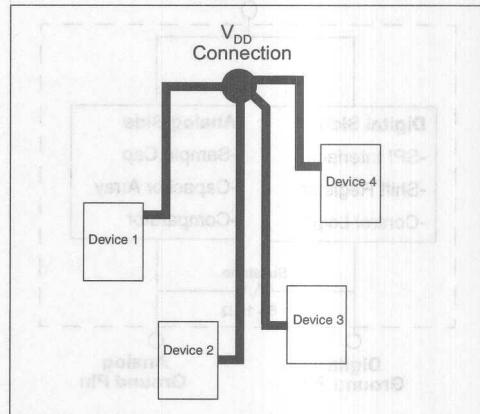


FIGURE 6-4: V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.

FilterLab is a trademark of Microchip Technology Inc. in the U.S.A and other countries. All rights reserved.

MCP3004/3008

6.5 Utilizing the Digital and Analog Ground Pins

The MCP3004/3008 devices provide both digital and analog ground connections to provide another means of noise reduction. As shown in Figure 6-5, the analog and digital circuitry is separated internal to the device. This reduces noise from the digital portion of the device being coupled into the analog portion of the device. The two grounds are connected internally through the substrate which has a resistance of 5 -10 Ω .

If no ground plane is utilized, then both grounds must be connected to V_{SS} on the board. If a ground plane is available, both digital and analog ground pins should be connected to the analog ground plane. If both an analog and a digital ground plane are available, both the digital and the analog ground pins should be connected to the analog ground plane. Following these steps will reduce the amount of digital noise from the rest of the board being coupled into the A/D Converter.

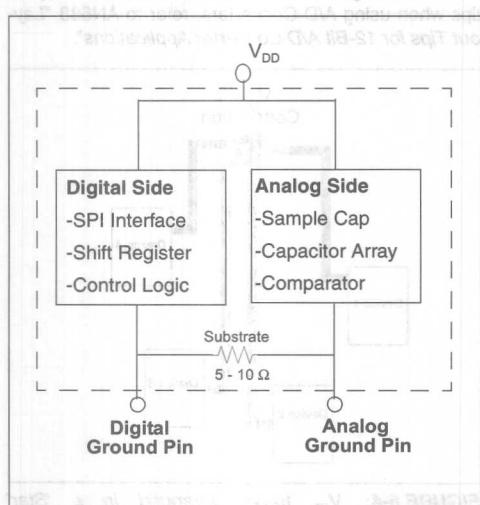
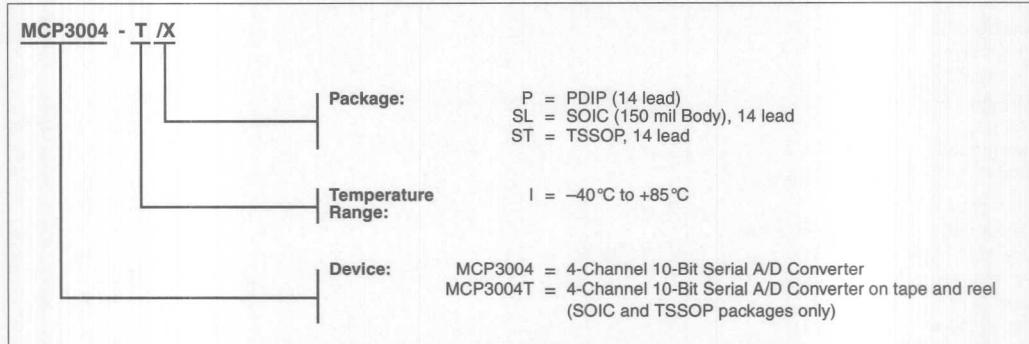


FIGURE 6-5: Separation of Analog and Digital Ground Pins

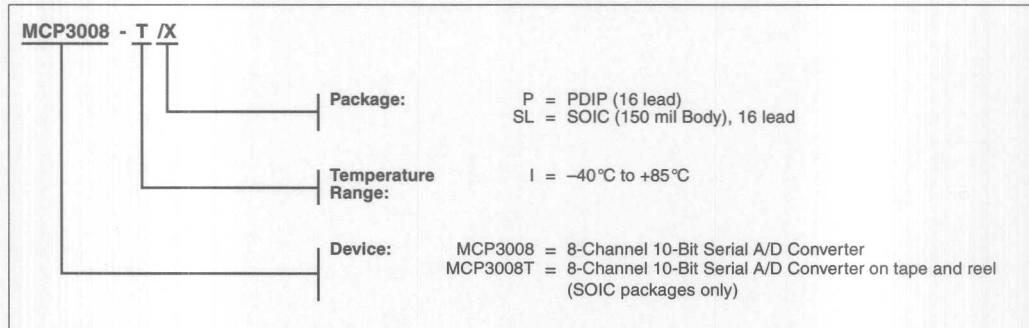
MCP3004 PRODUCT IDENTIFICATION SYSTEMS

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



MCP3008 PRODUCT IDENTIFICATION SYSTEMS

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Sales and Support

Data Sheets

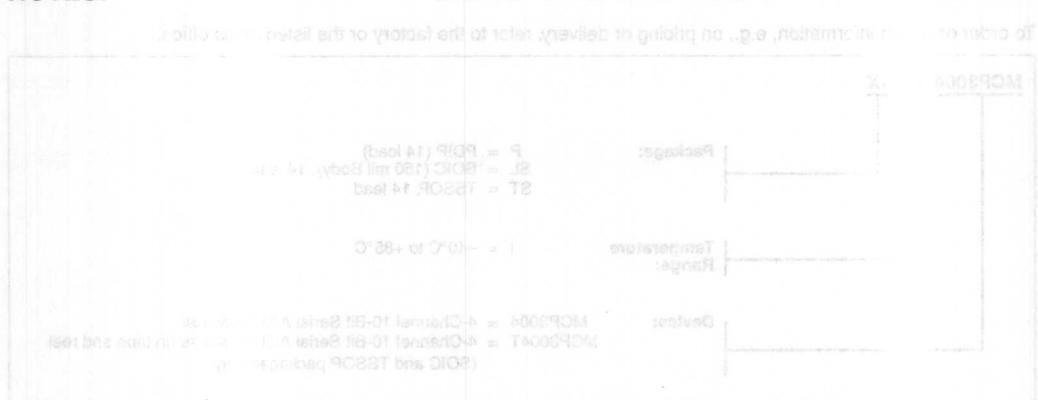
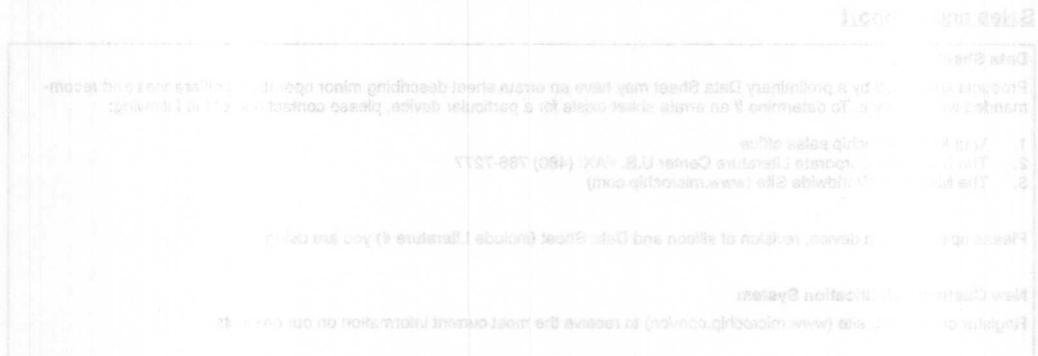
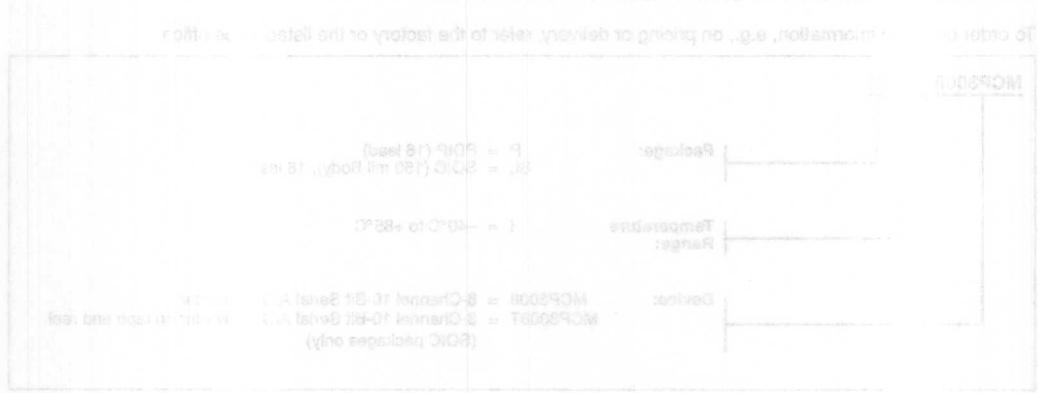
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

NOTES:**MCP2000 PRODUCT IDENTIFICATION SYSTEMS****MCP2000 PRODUCT IDENTIFICATION SYSTEMS**



MICROCHIP

MCP3201

2.7V 12-Bit A/D Converter with SPI™ Serial Interface

FEATURES

- 12-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL (MCP3201-B)
- ± 2 LSB max INL (MCP3201-C)
- On-chip sample and hold
- SPI™ serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 100ksps max. sampling rate at $V_{DD} = 5V$
- 50ksps max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
 - 500nA typical standby current, 2 μ A max.
 - 400 μ A max. active current at 5V
- Industrial temp range: -40°C to +85°C
- 8-pin PDIP, SOIC and TSSOP packages

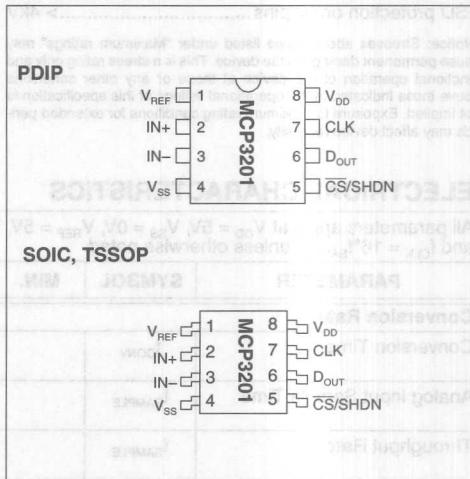
APPLICATIONS

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

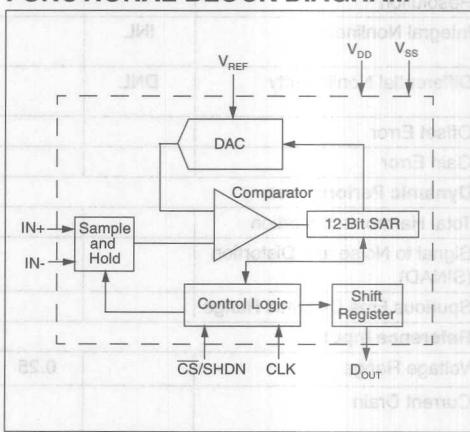
DESCRIPTION

The Microchip Technology Inc. MCP3201 is a successive approximation 12-bit Analog-to-Digital (A/D) Converter with on-board sample and hold circuitry. The device provides a single pseudo-differential input. Differential Nonlinearity (DNL) is specified at ± 1 LSB, and Integral Nonlinearity (INL) is offered in ± 1 LSB (MCP3201-B) and ± 2 LSB (MCP3201-C) versions. Communication with the device is done using a simple serial interface compatible with the SPI protocol. The device is capable of sample rates of up to 100ksps at a clock rate of 1.6MHz. The MCP3201 operates over a broad voltage range (2.7V - 5.5V). Low current design permits operation with typical standby and active currents of only 500nA and 300 μ A, respectively. The device is offered in 8-pin PDIP, TSSOP and 150mil SOIC packages.

PACKAGE TYPES



FUNCTIONAL BLOCK DIAGRAM



SPI is a trademark of Motorola Inc.

MCP3201

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to V_{DD} +0.6V
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	> 4kV

*Notice: Stresses above those listed under "Maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
V_{DD}	+2.7V to 5.5V Power Supply
V_{SS}	Ground
IN+	Positive Analog Input
IN-	Negative Analog Input
CLK	Serial Clock
D_{OUT}	Serial Data Out
CS/SHDN	Chip select/Shutdown Input
V_{REF}	Reference Voltage Input

ELECTRICAL CHARACTERISTICS

All parameters apply at $V_{DD} = 5V$, $V_{SS} = 0V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $f_{SAMPLE} = 100\text{ksps}$ and $f_{CLK} = 16 \cdot f_{SAMPLE}$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Conversion Rate						
Conversion Time	t_{CONV}			12	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}			100 50	ksps ksps	$V_{DD} = V_{REF} = 5V$ $V_{DD} = V_{REF} = 2.7V$
DC Accuracy						
Resolution			12		bits	
Integral Nonlinearity	INL		± 0.75 ± 1	± 1 ± 2	LSB LSB	MCP3201-B MCP3201-C
Differential Nonlinearity	DNL		± 0.5	± 1	LSB	No missing codes over temperature
Offset Error			± 1.25	± 3	LSB	
Gain Error			± 1.25	± 5	LSB	
Dynamic Performance						
Total Harmonic Distortion			-82		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Signal to Noise and Distortion (SINAD)			72		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Spurious Free Dynamic Range			86		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Reference Input						
Voltage Range		0.25		V_{DD}	V	Note 2
Current Drain		100 .001	150 3	μA	μA	$CS = V_{DD} = 5V$
Analog Inputs						
Input Voltage Range (IN+)		IN-		$V_{REF} + IN -$	V	
Input Voltage Range (IN-)		$V_{SS} - 100$		$V_{SS} + 100$	mV	
Leakage Current			0.001	± 1	μA	
Switch Resistance	R_{SS}		1K		Ω	See Figure 4-1
Sample Capacitor	C_{SAMPLE}		20		pF	See Figure 4-1

ELECTRICAL CHARACTERISTICS (CONTINUED)

All parameters apply at $V_{DD} = 5V$, $V_{SS} = 0V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 100\text{ksps}$ and $f_{CLK} = 16 \cdot f_{SAMPLE}$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Digital Input/Output						
Data Coding Format			Straight Binary			
High Level Input Voltage	V_{IH}	$0.7 V_{DD}$			V	
Low Level Input Voltage	V_{IL}			$0.3 V_{DD}$	V	
High Level Output Voltage	V_{OH}	4.1			V	$I_{OH} = -1\text{mA}$, $V_{DD} = 4.5\text{V}$
Low Level Output Voltage	V_{OL}			0.4	V	$I_{OL} = 1\text{mA}$, $V_{DD} = 4.5\text{V}$
Input Leakage Current	I_{IU}	-10		10	μA	$V_{IN} = V_{SS}$ or V_{DD}
Output Leakage Current	I_{LO}	-10		10	μA	$V_{OUT} = V_{SS}$ or V_{DD}
Pin Capacitance (all inputs/outputs)	C_{IN} , C_{OUT}			10	pF	$V_{DD} = 5.0\text{V}$ (Note 1) $T_{AMB} = 25^{\circ}\text{C}$, $f = 1\text{MHz}$
Timing Parameters						
Clock Frequency	f_{CLK}			1.6 0.8	MHz MHz	$V_{DD} = 5\text{V}$ (Note 3) $V_{DD} = 2.7\text{V}$ (Note 3)
Clock High Time	t_{HI}	312			ns	
Clock Low Time	t_{LO}	312			ns	
\bar{CS} Fall To First Rising CLK Edge	t_{SUCS}	100			ns	
CLK Fall To Output Data Valid	t_{DO}			200	ns	See Test Circuits, Figure 1-2
CLK Fall To Output Enable	t_{EN}			200	ns	See Test Circuits, Figure 1-2
\bar{CS} Rise To Output Disable	t_{DIS}			100	ns	See Test Circuits, Figure 1-2 (Note 1)
\bar{CS} Disable Time	t_{CSH}	625			ns	
D_{OUT} Rise Time	t_R			100	ns	See Test Circuits, Figure 1-2 (Note 1)
D_{OUT} Fall Time	t_F			100	ns	See Test Circuits, Figure 1-2 (Note 1)
Power Requirements						
Operating Voltage	V_{DD}	2.7		5.5	V	
Operating Current	I_{DD}		300 210	400	μA μA	$V_{DD} = 5.0\text{V}$, D_{OUT} unloaded $V_{DD} = 2.7\text{V}$, D_{OUT} unloaded
Standby Current	I_{DDS}		0.5	2	μA	$\bar{CS} = V_{DD} = 5.0\text{V}$

Note 1: This parameter is guaranteed by characterization and not 100% tested.

2: See graph that relates linearity performance to V_{REF} level.

3: Because the sample cap will eventually lose charge, effective clock rates below 10kHz can affect linearity performance, especially at elevated temperatures. See Section 6.2 for more information.

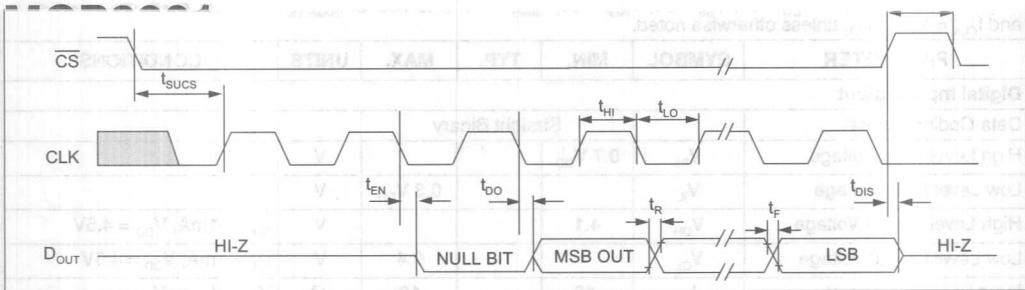


FIGURE 1-1: Serial Timing.

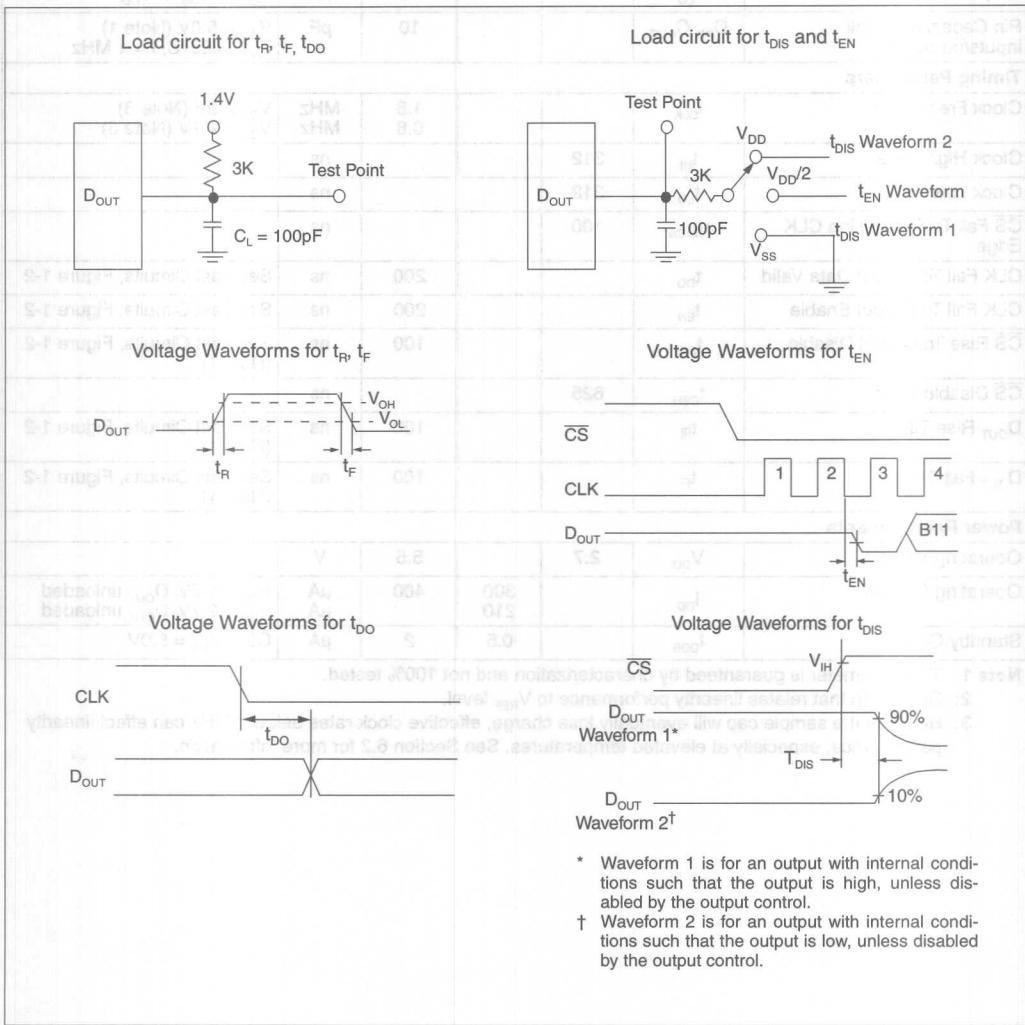


FIGURE 1-2: Test Circuits.

2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

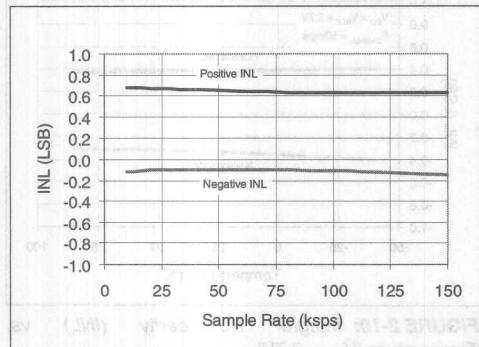


FIGURE 2-1: Integral Nonlinearity (INL) vs. Sample Rate.

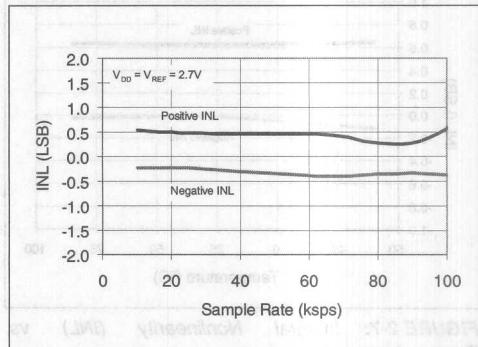


FIGURE 2-4: Integral Nonlinearity (INL) vs. Sample Rate ($V_{DD} = 2.7V$).

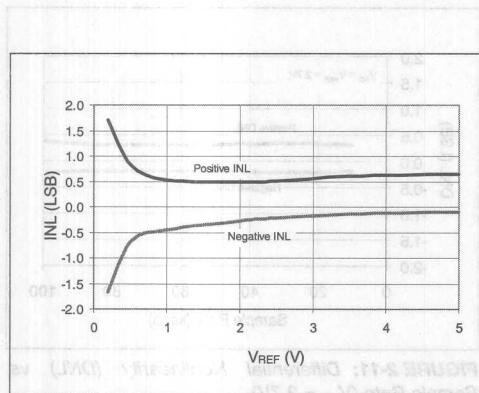


FIGURE 2-2: Integral Nonlinearity (INL) vs. V_{REF} .

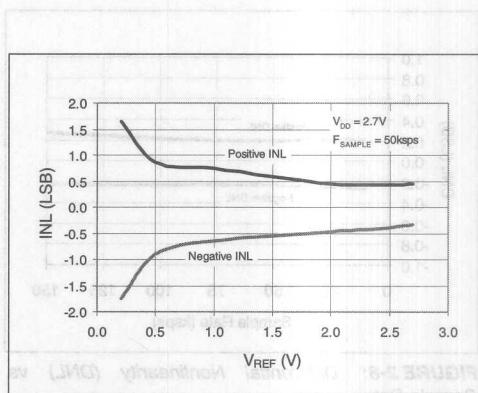


FIGURE 2-5: Integral Nonlinearity (INL) vs. V_{REF} ($V_{DD} = 2.7V$).

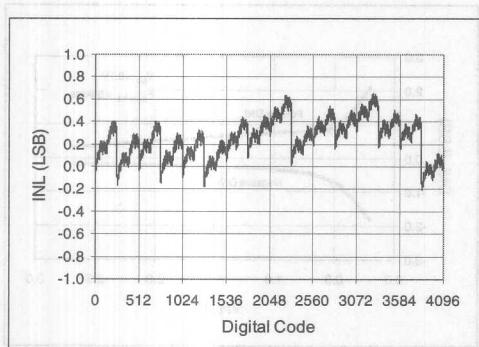


FIGURE 2-3: Integral Nonlinearity (INL) vs. Code (Representative Part).

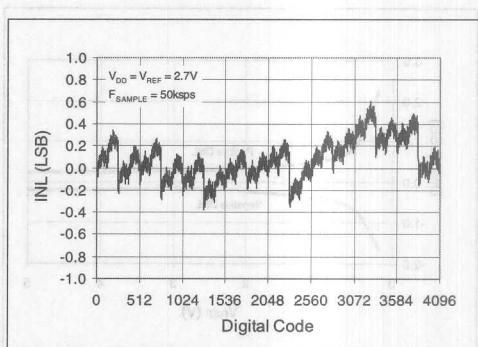


FIGURE 2-6: Integral Nonlinearity (INL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

MCP3201

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

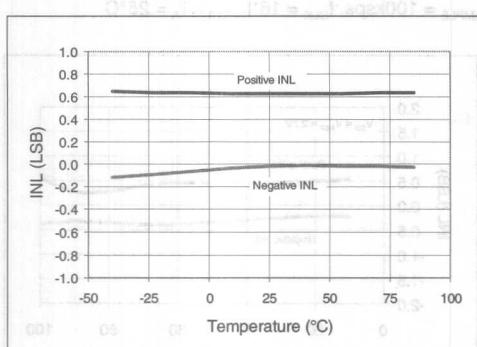


FIGURE 2-7: Integral Nonlinearity (INL) vs. Temperature.

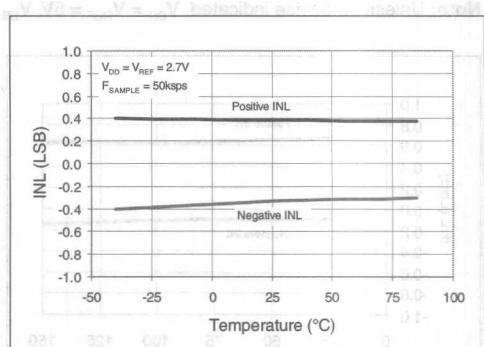


FIGURE 2-10: Integral Nonlinearity (INL) vs. Temperature ($V_{DD} = 2.7V$).

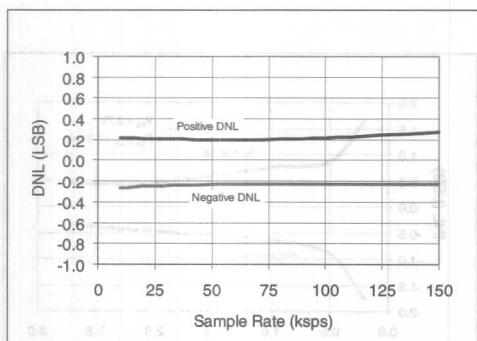


FIGURE 2-8: Differential Nonlinearity (DNL) vs. Sample Rate.

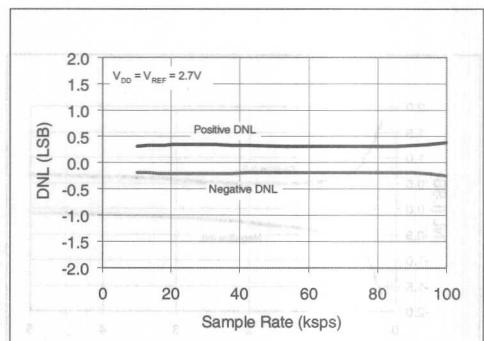


FIGURE 2-11: Differential Nonlinearity (DNL) vs. Sample Rate ($V_{DD} = 2.7V$).

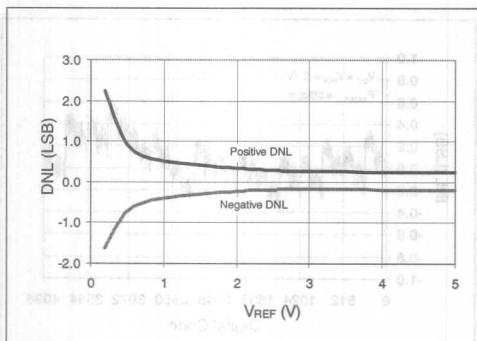


FIGURE 2-9: Differential Nonlinearity (DNL) vs. V_{REF} .

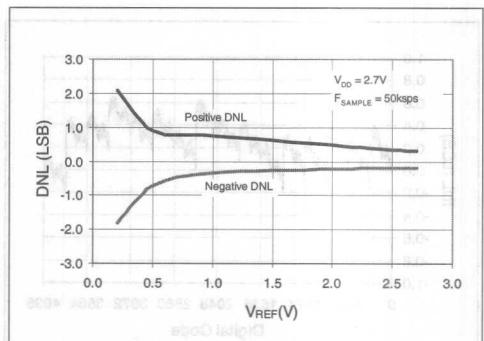


FIGURE 2-12: Differential Nonlinearity (DNL) vs. V_{REF} ($V_{DD} = 2.7V$).

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 16 \cdot f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

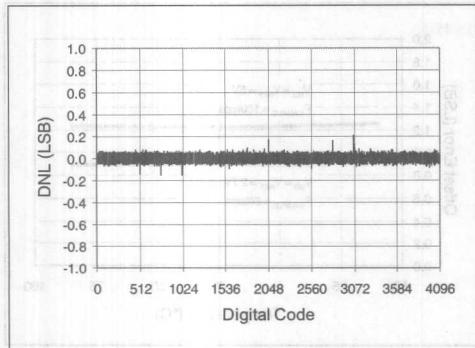


FIGURE 2-13: Differential Nonlinearity (DNL) vs. Code (Representative Part).

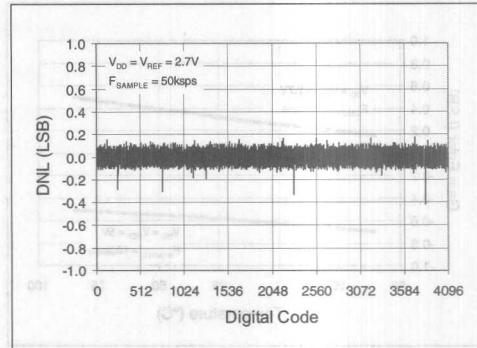


FIGURE 2-16: Differential Nonlinearity (DNL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

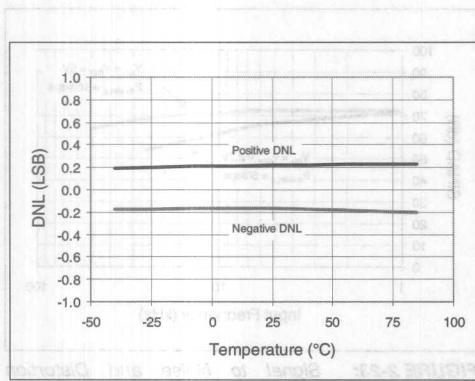


FIGURE 2-14: Differential Nonlinearity (DNL) vs. Temperature.

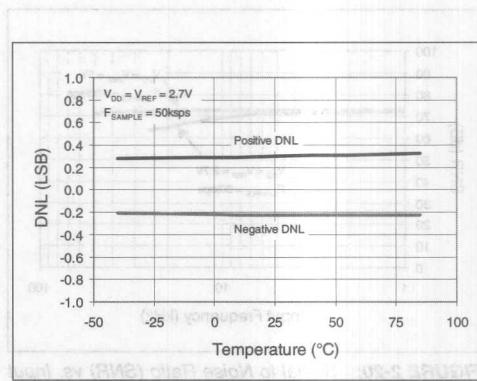


FIGURE 2-17: Differential Nonlinearity (DNL) vs. Temperature ($V_{DD} = 2.7V$).

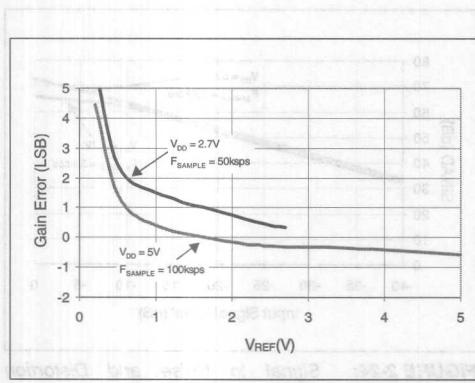


FIGURE 2-15: Gain Error vs. V_{REF} .

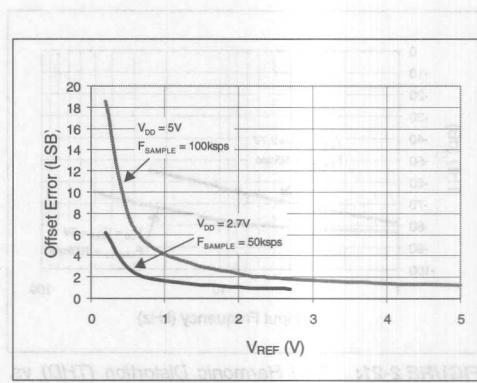


FIGURE 2-18: Offset Error vs. V_{REF} .

MCP3201

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

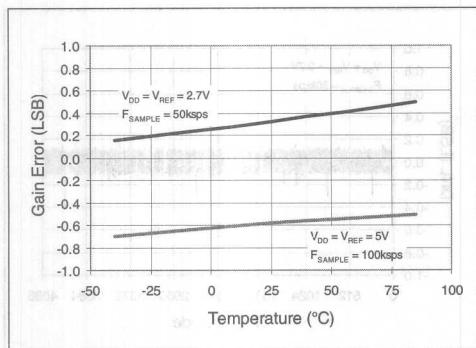


FIGURE 2-19: Gain Error vs. Temperature.

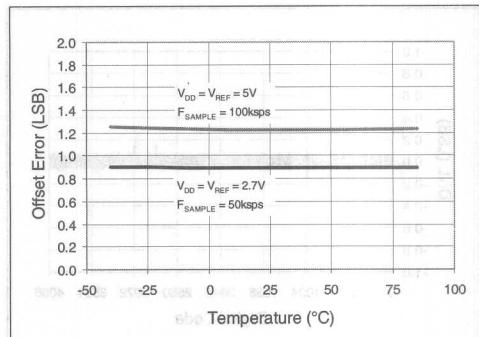


FIGURE 2-22: Offset Error vs. Temperature.

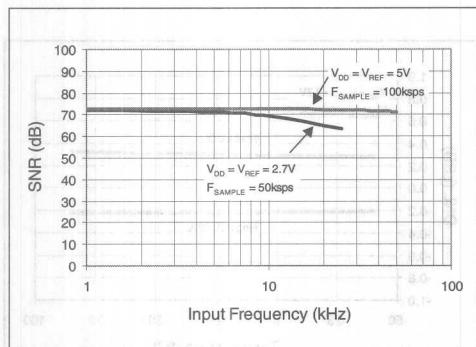


FIGURE 2-20: Signal to Noise Ratio (SNR) vs. Input Frequency.

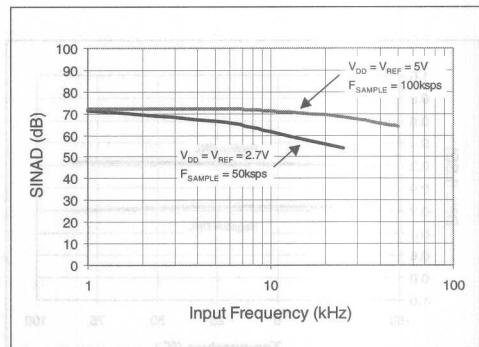


FIGURE 2-23: Signal to Noise and Distortion (SINAD) vs. Input Frequency.

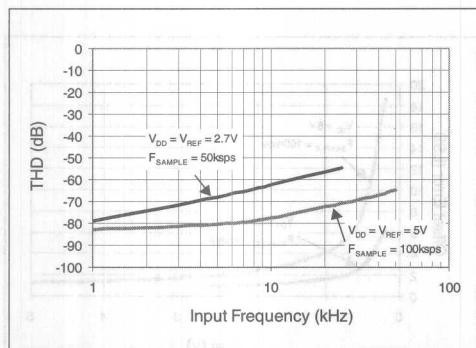


FIGURE 2-21: Total Harmonic Distortion (THD) vs. Input Frequency.

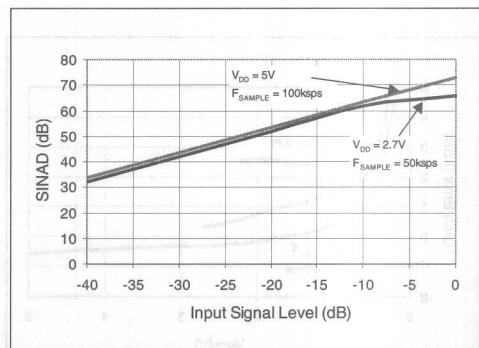


FIGURE 2-24: Signal to Noise and Distortion (SINAD) vs. Input Signal Level.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{kspS}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

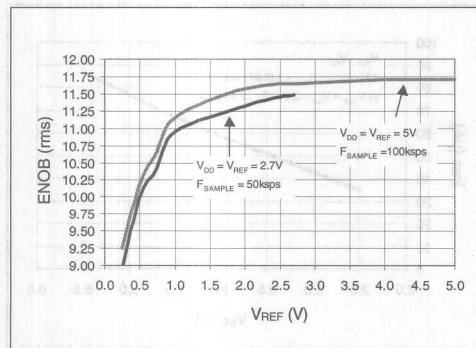


FIGURE 2-25: Effective Number of Bits (ENOB) vs. V_{REF} .

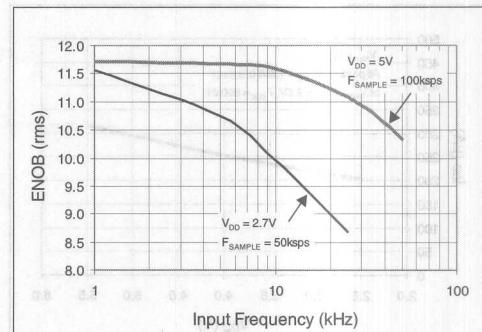


FIGURE 2-28: Effective Number of Bits (ENOB) vs. Input Frequency.

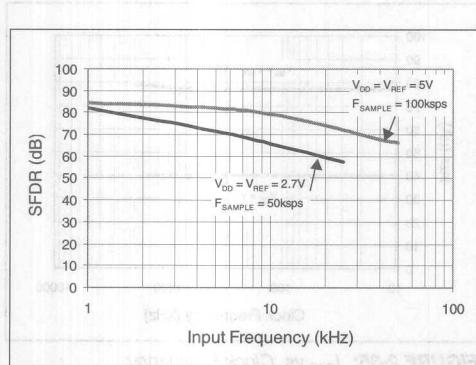


FIGURE 2-26: Spurious Free Dynamic Range (SFDR) vs. Input Frequency.

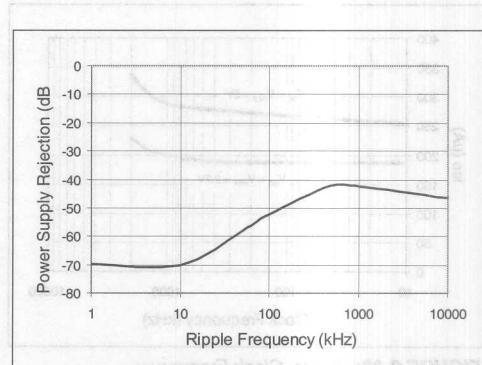


FIGURE 2-29: Power Supply Rejection (PSR) vs. Ripple Frequency.

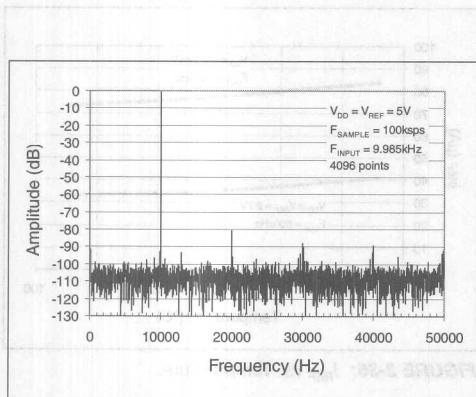


FIGURE 2-27: Frequency Spectrum of 10kHz input (Representative Part).

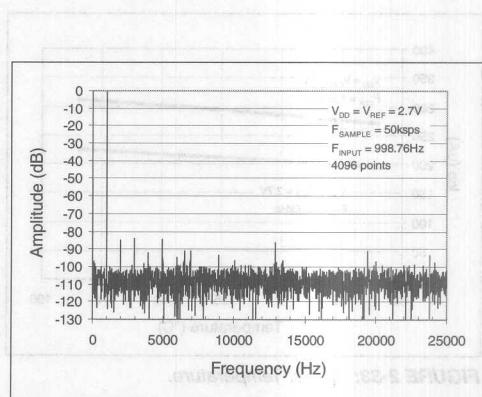


FIGURE 2-30: Frequency Spectrum of 1kHz input (Representative Part), $V_{DD} = 2.7V$.

MCP3201

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 16f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

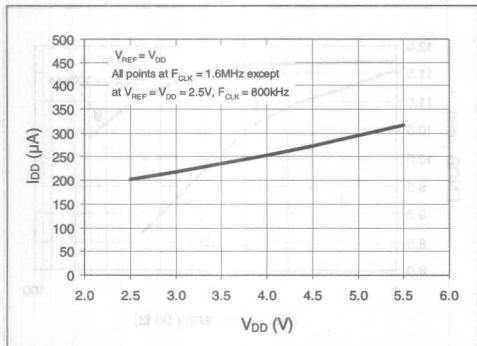


FIGURE 2-31: I_{DD} vs. V_{DD} .

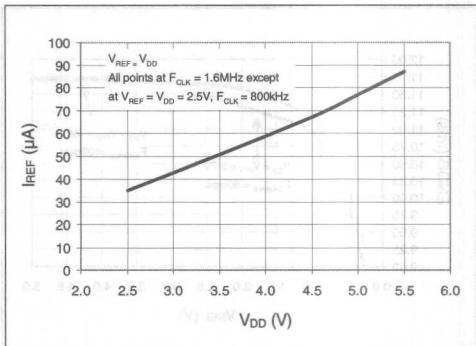


FIGURE 2-34: I_{REF} vs. V_{DD} .

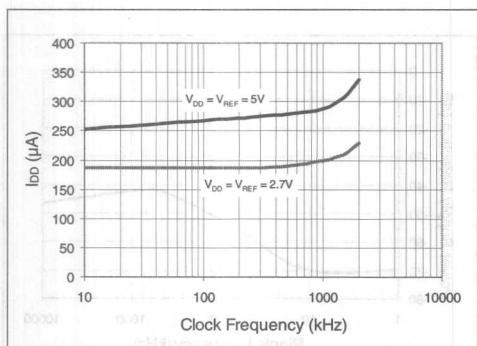


FIGURE 2-32: I_{DD} vs. Clock Frequency.

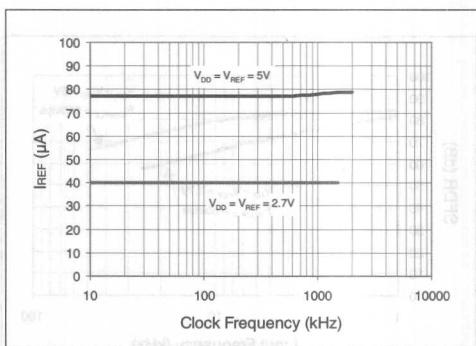


FIGURE 2-35: I_{REF} vs. Clock Frequency.

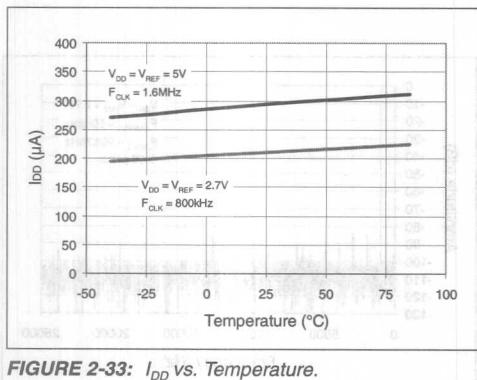


FIGURE 2-33: I_{DD} vs. Temperature.

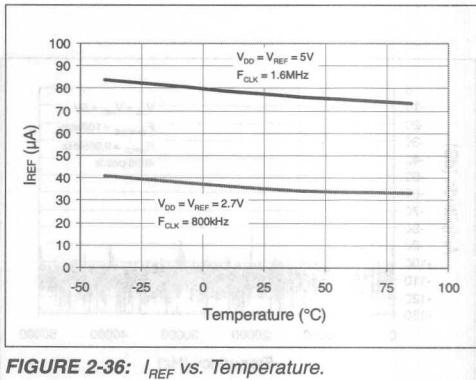


FIGURE 2-36: I_{REF} vs. Temperature.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 16 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

Output (I_{DOS}) is defined as the current sink at pins 18 & 19.

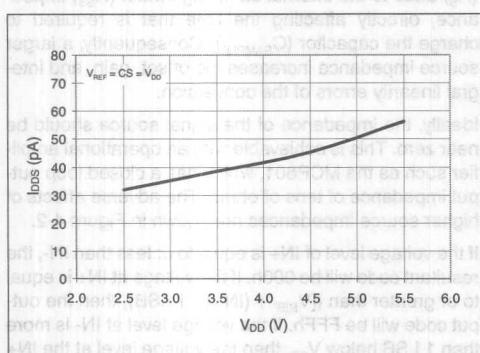


FIGURE 2-37: I_{DOS} vs. V_{DD} .

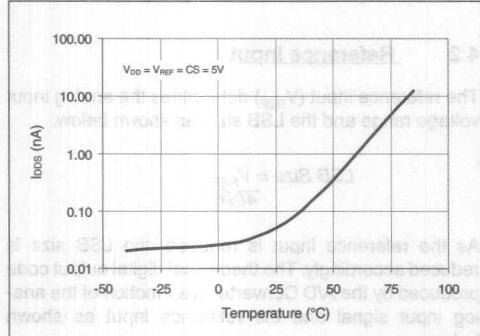


FIGURE 2-38: I_{DOS} vs. Temperature.

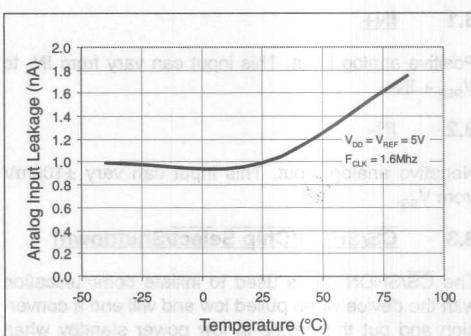


FIGURE 2-39: Analog Input Leakage Current vs. Temperature.

3.1 IN+

Positive analog input. This input can vary from IN- to $V_{REF} + IN_-$.

3.2 IN-

Negative analog input. This input can vary $\pm 100mV$ from V_{SS} .

3.3 CS/SHDN(Chip Select/Shutdown)

The CS/SHDN pin is used to initiate communication with the device when pulled low and will end a conversion and put the device in low power standby when pulled high. The CS/SHDN pin must be pulled high between conversions.

3.4 CLK (Serial Clock)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 6.2 for constraints on clock speed.

3.5 DOUT (Serial Data output)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

4.0 DEVICE OPERATION

The MCP3201 A/D Converter employs a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the first rising edge of the serial clock after CS has been pulled low. Following this sample time, the input switch of the converter opens and the device uses the collected charge on the internal sample and hold capacitor to produce a serial 12-bit digital output code. Conversion rates of 100ksps are possible on the MCP3201. See Section 6.2 for information on minimum clock rates. Communication with the device is done using a 3-wire SPI-compatible interface.

4.1 Analog Inputs

The MCP3201 provides a single pseudo-differential input. The IN+ input can range from IN- to V_{REF} ($V_{REF} + IN_-$). The IN- input is limited to $\pm 100mV$ from the V_{SS} rail. The IN- input can be used to cancel small signal common-mode noise which is present on both the IN+ and IN- inputs.

For the A/D Converter to meet specification, the charge holding capacitor (C_{SAMPLE}) must be given enough time to acquire a 12-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

(R_S) adds to the internal sampling switch (R_{SS}) impedance, directly affecting the time that is required to charge the capacitor (C_{SAMPLE}). Consequently, a larger source impedance increases the offset, gain, and integral linearity errors of the conversion.

Ideally, the impedance of the signal source should be near zero. This is achievable with an operational amplifier such as the MCP601, which has a closed loop output impedance of tens of ohms. The adverse affects of higher source impedances are shown in Figure 4-2.

If the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h. If the voltage at IN+ is equal to or greater than $\{[V_{REF} + (IN_-)] - 1 \text{ LSB}\}$, then the output code will be FFFh. If the voltage level at IN- is more than 1 LSB below V_{SS} , then the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS} , then the FFFh code will not be seen unless the IN+ input goes above V_{REF} level.

4.2 Reference Input

The reference input (V_{REF}) determines the analog input voltage range and the LSB size, as shown below.

$$\text{LSB Size} = \frac{V_{REF}}{4096}$$

As the reference input is reduced, the LSB size is reduced accordingly. The theoretical digital output code produced by the A/D Converter is a function of the analog input signal and the reference input as shown below.

$$\text{Digital Output Code} = \frac{4096 * V_{IN}}{V_{REF}}$$

where:

$$V_{IN} = \text{analog input voltage} = V(IN+) - V(IN-)$$

$$V_{REF} = \text{reference voltage}$$

When using an external voltage reference device, the system designer should always refer to the manufacturer's recommendations for circuit layout. Any instability in the operation of the reference device will have a direct effect on the operation of the A/D Converter.

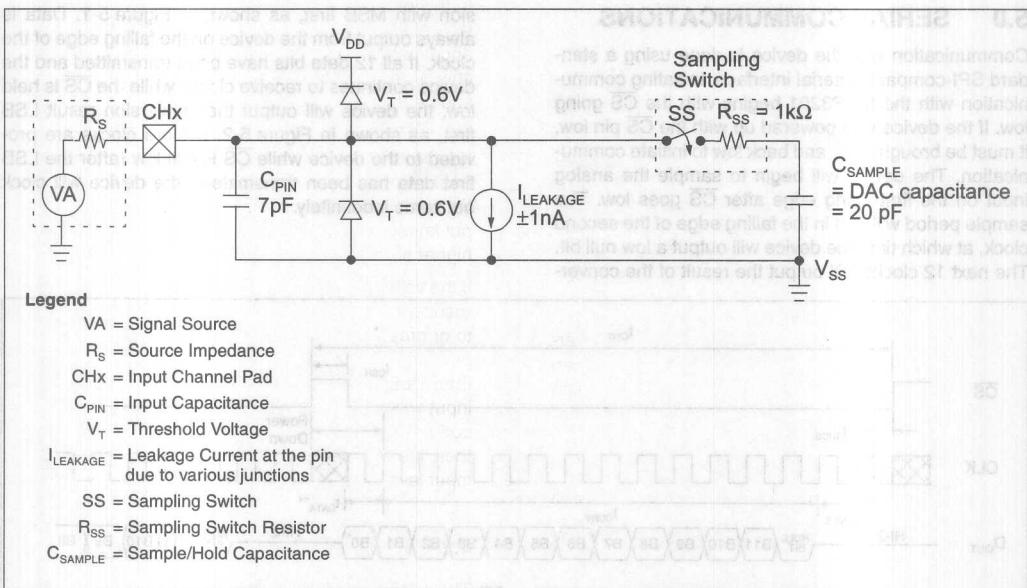


FIGURE 4-1: Analog Input Model.

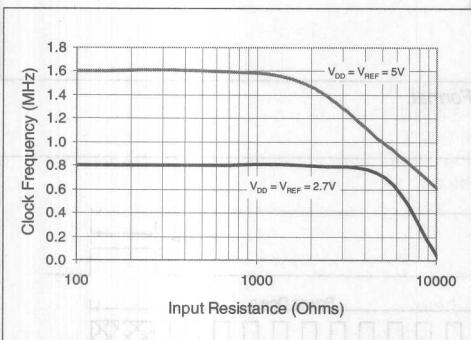


FIGURE 4-2: Maximum Clock Frequency vs. Input Resistance (R_S) to maintain less than a 0.1 LSB deviation in INL from nominal conditions.

5.0 SERIAL COMMUNICATIONS

Communication with the device is done using a standard SPI-compatible serial interface. Initiating communication with the MCP3201 begins with the CS going low. If the device was powered up with the CS pin low, it must be brought high and back low to initiate communication. The device will begin to sample the analog input on the first rising edge after CS goes low. The sample period will end in the falling edge of the second clock, at which time the device will output a low null bit. The next 12 clocks will output the result of the conver-

sion with MSB first, as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 12 data bits have been transmitted and the device continues to receive clocks while the CS is held low, the device will output the conversion result LSB first, as shown in Figure 5-2. If more clocks are provided to the device while CS is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

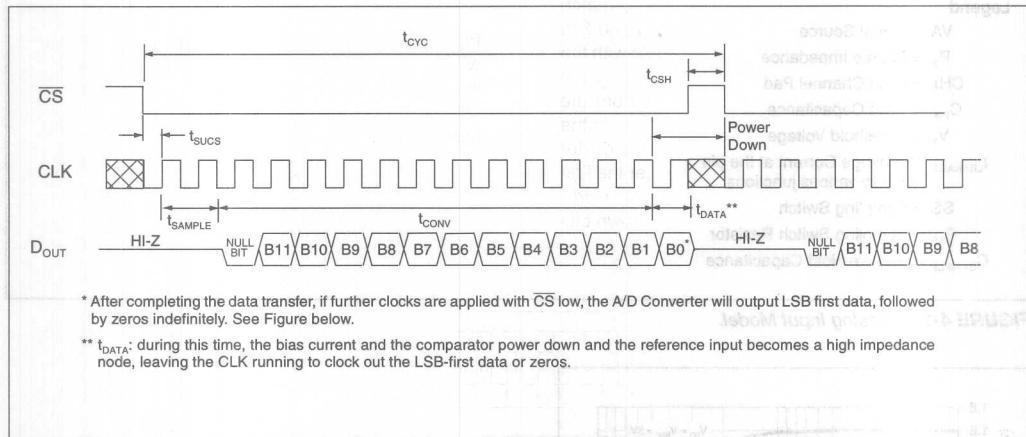


FIGURE 5-1: Communication with MCP3201 using MSB first Format.

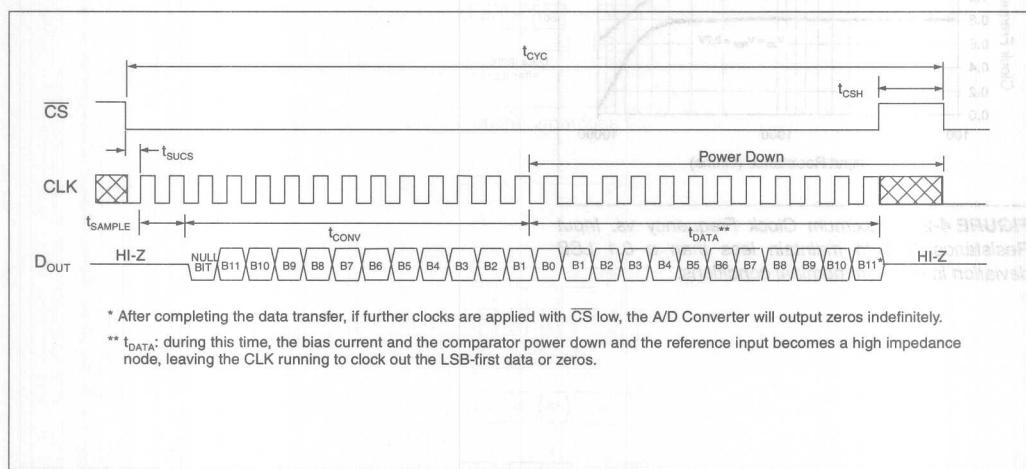


FIGURE 5-2: Communication with MCP3201 using LSB first Format.

6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3201 with Microcontroller SPI Ports

With most microcontroller SPI ports, it is required to clock out eight bits at a time. If this is the case, it will be necessary to provide more clocks than are required for the MCP3201. As an example, Figure 6-1 and Figure 6-2 show how the MCP3201 can be interfaced to a microcontroller with a standard SPI port. Since the MCP3201 always clocks data out on the falling edge of clock, the MCU SPI port must be configured to match this operation. SPI Mode 0,0 (clock idles low) and SPI Mode 1,1 (clock idles high) are both compatible with the MCP3201. Figure 6-1 depicts the operation shown in SPI Mode 0,0, which requires that the CLK from the microcontroller idles in the 'low' state. As shown in the diagram, the MSB is clocked out of the A/D Converter on the falling edge of the third clock pulse. After the first eight clocks have been sent to the device, the microcontroller's receive buffer will contain two unknown bits

(the output is at high impedance for the first two clocks), the null bit and the highest order five bits of the conversion. After the second eight clocks have been sent to the device, the MCU receive register will contain the lowest order seven bits and the B1 bit repeated as the A/D Converter has begun to shift out LSB first data with the extra clock. Typical procedure would then call for the lower order byte of data to be shifted right by one bit to remove the extra B1 bit. The B7 bit is then transferred from the high order byte to the lower order byte, and then the higher order byte is shifted one bit to the right as well. Easier manipulation of the converted data can be obtained by using this method.

Figure 6-2 shows the same thing in SPI Mode 1,1 which requires that the clock idles in the high state. As with mode 0,0, the A/D Converter outputs data on the falling edge of the clock and the MCU latches data from the A/D Converter on the rising edge of the clock.

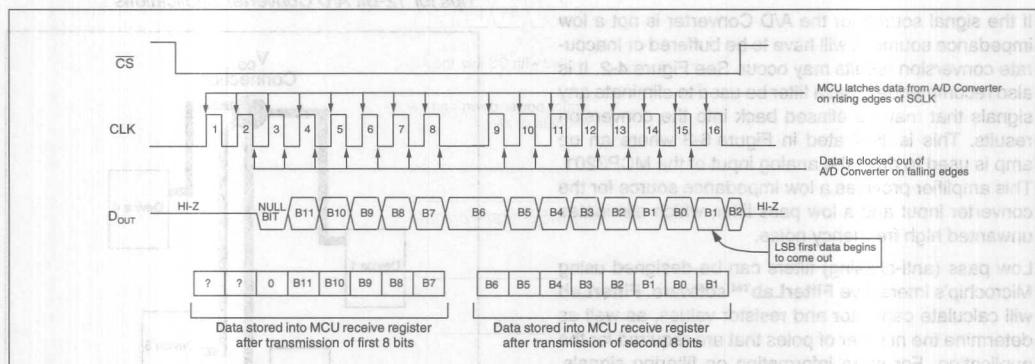


FIGURE 6-1: SPI Communication using 8-bit segments (Mode 0,0: SCLK idles low).

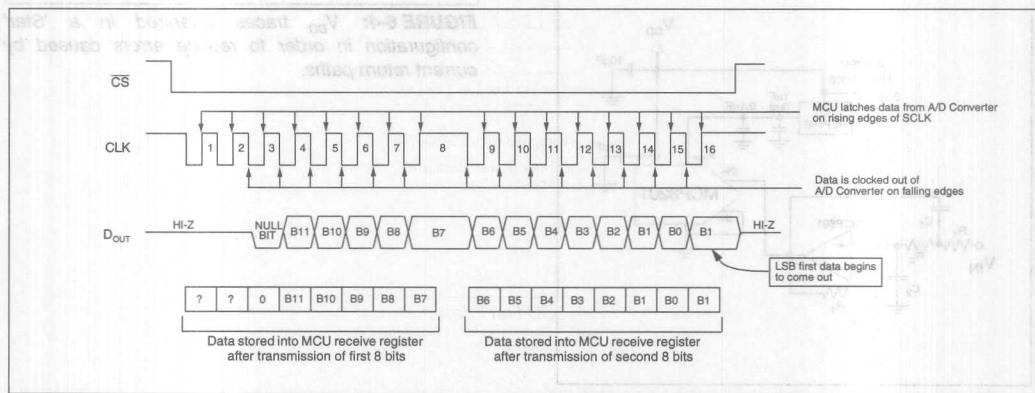


FIGURE 6-2: SPI Communication using 8-bit segments (Mode 1,1: SCLK idles high).

6.2 Maintaining Minimum Clock Speed

When the MCP3201 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample cap while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample capacitor for at least 1.2ms after the sample period has ended. This means that the time between the end of the sample period and the time that all 12 data bits have been clocked out must not exceed 1.2ms (effective clock frequency of 10kHz). Failure to meet this criteria may induce linearity errors into the conversion outside the rated specifications. It should be noted that during the entire conversion cycle, the A/D Converter does not require a constant clock speed or duty cycle, as long as all timing specifications are met.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the A/D Converter is not a low impedance source, it will have to be buffered or inaccurate conversion results may occur. See Figure 6-2. It is also recommended that a filter be used to eliminate any signals that may be aliased back into the conversion results. This is illustrated in Figure 6-3 where an op amp is used to drive the analog input of the MCP3201. This amplifier provides a low impedance source for the converter input and a low pass filter, which eliminates unwanted high frequency noise.

Low pass (anti-aliasing) filters can be designed using Microchip's interactive **FilterLab™** software. **FilterLab** will calculate capacitor and resistor values, as well as determine the number of poles that are required for the application. For more information on filtering signals, see the application note AN699 "Anti-Aliasing Analog Filters for Data Acquisition Systems."

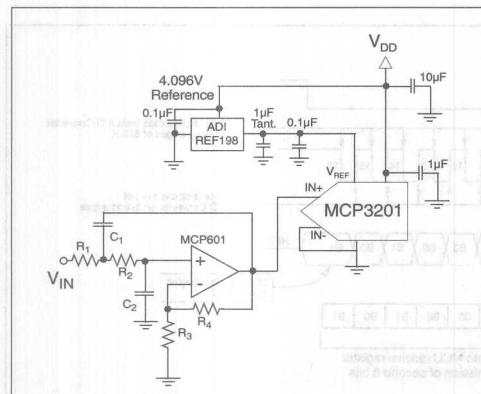


FIGURE 6-3: The MCP601 Operational Amplifier is used to implement a 2nd order anti-aliasing filter for the signal being converted by the MCP3201.

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of 1 μ F is recommended.

Digital and analog traces should be separated as much as possible on the board and no traces should run underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating current return paths and associated errors. See Figure 6-4. For more information on layout tips when using A/D Converter, refer to AN688 "Layout Tips for 12-Bit A/D Converter Applications".

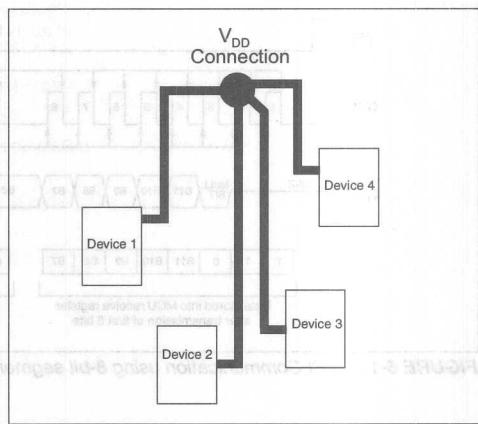
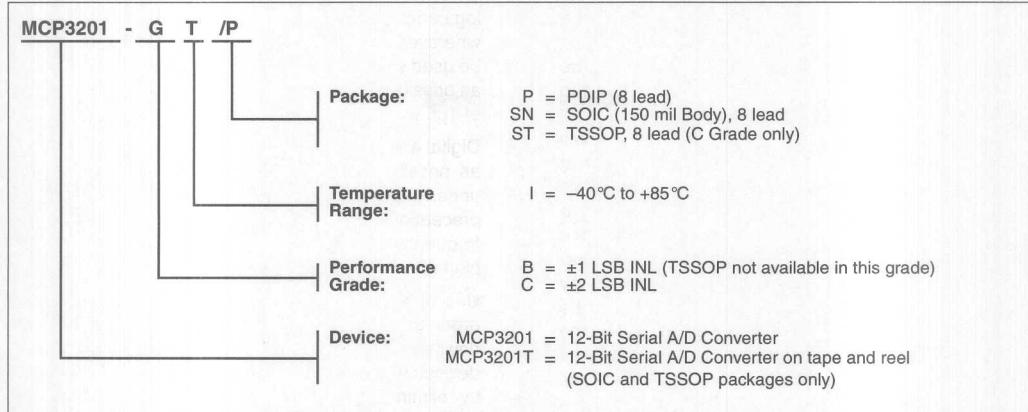


FIGURE 6-4: V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.

FilterLab is a trademark of Microchip Technology Inc. in the U.S.A and other countries. All rights reserved.

MCP3201 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277. After September 1, 1999, (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCP3201

NOTES:

MICROCHIP TOUCHSCREEN SYSTEM

To obtain detailed information, e.g., on pinout or detailed interface to the host, refer to the appropriate data sheet.

Pin	Name	Function	Pin	Name	Function
9	PGND (G) Input		T	TD	PGND (G)
8	SI (S) Input		2	RD	PGND (G)
7	SD (D) Input		3	WR	PGND (G)
6	CS (C) Input		4	DATA	PGND (G)
5	RDY (R) Output		5	CS	PGND (G)
4	WRY (W) Output		6	RDY	PGND (G)
3	DATA	PGND (G)	7	WR	PGND (G)
2	RD	PGND (G)	8	SI	PGND (G)
1	TD	PGND (G)	9	PGND (G)	PGND (G)

Pin	Name	Description
9	PGND (G)	System ground connection
8	SI (S)	Serial input
7	SD (D)	Serial data output
6	CS (C)	Chip select
5	RDY (R)	Ready output
4	WRY (W)	Write enable
3	DATA	Parallel data output
2	RD	Read enable
1	TD	Parallel data input



MCP3202

2.7V Dual Channel 12-Bit A/D Converter with SPI™ Serial Interface

FEATURES

- 12-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL (MCP3202-B)
- ± 2 LSB max INL (MCP3202-C)
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI™ serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 100ksps max. sampling rate at $V_{DD} = 5V$
- 50ksps max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
 - 500nA typical standby current, 5 μ A max.
 - 550 μ A max. active current at 5V
- Industrial temp range: -40°C to +85°C
- 8-pin PDIP SOIC and TSSOP packages

APPLICATIONS

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

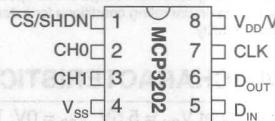
DESCRIPTION

The Microchip Technology Inc. MCP3202 is a successive approximation 12-bit Analog-to-Digital (A/D) Converter with on-board sample and hold circuitry. The MCP3202 is programmable to provide a single pseudo-differential input pair or dual single-ended inputs. Differential Nonlinearity (DNL) is specified at ± 1 LSB, and Integral Nonlinearity (INL) is offered in ± 1 LSB (MCP3202-B) and ± 2 LSB (MCP3202-C) versions. Communication with the device is done using a simple serial interface compatible with the SPI protocol. The device is capable of conversion rates of up to 100ksps at 5V and 50ksps at 2.7V. The MCP3202 device operates over a broad voltage range (2.7V - 5.5V). Low current design permits operation with typical standby and active currents of only 500nA and 375 μ A, respectively. The MCP3202 is offered in 8-pin PDIP, TSSOP and 150mil SOIC packages.

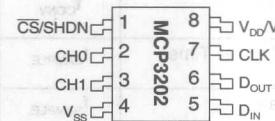
SPI is a trademark of Motorola Inc.

PACKAGE TYPES

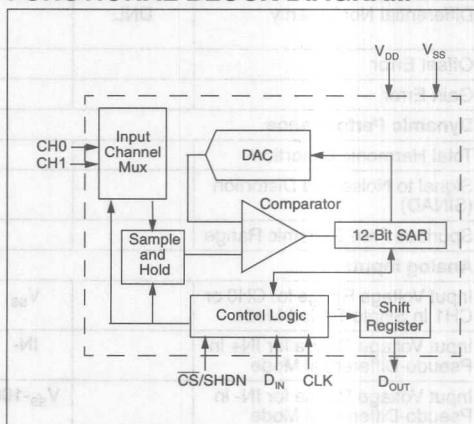
PDIP



SOIC, TSSOP



FUNCTIONAL BLOCK DIAGRAM



MCP3202

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD}	7.0V
All inputs and outputs w.r.t. V_{SS}	-0.6V to V_{DD} +0.6V
Storage temperature	-65°C to +150°C
Ambient temp. with power applied	-65°C to +125°C
Soldering temperature of leads (10 seconds)	+300°C
ESD protection on all pins	> 4kV

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
V_{DD}/V_{REF}	+2.7V to 5.5V Power Supply and Reference Voltage Input
CH0	Channel 0 Analog Input
CH1	Channel 1 Analog Input
CLK	Serial Clock
D _{IN}	Serial Data In
D _{OUT}	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input

ELECTRICAL CHARACTERISTICS

All parameters apply at $V_{DD} = 5.5V$, $V_{SS} = 0V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 100\text{ksps}$ and $f_{CLK} = 18*f_{SAMPLE}$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Conversion Rate						
Conversion Time	t_{CONV}			12	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}			100 50	ksps ksps	$V_{DD} = V_{REF} = 5V$ $V_{DD} = V_{REF} = 2.7V$
DC Accuracy						
Resolution			12		bits	
Integral Nonlinearity	INL		± 0.75 ± 1	± 1 ± 2	LSB LSB	MCP3202-B MCP3202-C
Differential Nonlinearity	DNL		± 0.5	± 1	LSB	No missing codes over temperature
Offset Error			± 1.25	± 3	LSB	
Gain Error			± 1.25	± 5	LSB	
Dynamic Performance						
Total Harmonic Distortion			-82		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Signal to Noise and Distortion (SINAD)			72		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Spurious Free Dynamic Range			86		dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Analog Inputs						
Input Voltage Range for CH0 or CH1 in Single-Ended Mode		V_{SS}		V_{REF}	V	
Input Voltage Range for IN+ in Pseudo-Differential Mode		IN-		$V_{REF}+IN-$		See Sections 3.1 and 4.1
Input Voltage Range for IN- in Pseudo-Differential Mode		$V_{SS}-100$		$V_{SS}+100$	mV	See Sections 3.1 and 4.1
Leakage Current			.001	± 1	μA	
Switch Resistance	R_{SS}		1K		Ω	See Figure 4-1
Sample Capacitor	C_{SAMPLE}		20		pF	See Figure 4-1

ELECTRICAL CHARACTERISTICS (CONTINUED)

All parameters apply at $V_{DD} = 5.5V$, $V_{SS} = 0V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 100\text{kspS}$ and $f_{CLK} = 18*f_{SAMPLE}$ unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Digital Input/Output						
Data Coding Format						
High Level Input Voltage	V_{IH}	0.7 V_{DD}			V	
Low Level Input Voltage	V_{IL}			0.3 V_{DD}	V	
High Level Output Voltage	V_{OH}	4.1			V	$I_{OH} = -1\text{mA}$, $V_{DD} = 4.5\text{V}$
Low Level Output Voltage	V_{OL}			0.4	V	$I_{OL} = 1\text{mA}$, $V_{DD} = 4.5\text{V}$
Input Leakage Current	I_{LI}	-10		10	μA	$V_{IN} = V_{SS}$ or V_{DD}
Output Leakage Current	I_{LO}	-10		10	μA	$V_{OUT} = V_{SS}$ or V_{DD}
Pin Capacitance (All Inputs/Outputs)	C_{IN}, C_{OUT}			10	pF	$V_{DD} = 5.0\text{V}$ (Note 1) $T_{AMB} = 25^{\circ}\text{C}$, $f = 1\text{MHz}$
Timing Parameters						
Clock Frequency	f_{CLK}			1.8 0.9	MHz MHz	$V_{DD} = 5\text{V}$ (Note 2) $V_{DD} = 2.7\text{V}$ (Note 2)
Clock High Time	t_{HI}	250			ns	
Clock Low Time	t_{LO}	250			ns	
CS Fall To First Rising CLK Edge	t_{SUCS}	100			ns	
Data Input Setup Time	t_{SU}			50	ns	
Data Input Hold Time	t_{HD}			50	ns	
CLK Fall To Output Data Valid	t_{DO}			200	ns	See Test Circuits, Figure 1-2
CLK Fall To Output Enable	t_{EN}			200	ns	See Test Circuits, Figure 1-2
CS Rise To Output Disable	t_{DIS}			100	ns	See Test Circuits, Figure 1-2 Note 1
CS Disable Time	t_{CSH}	500			ns	
D_{OUT} Rise Time	t_R			100	ns	See Test Circuits, Figure 1-2 Note 1
D_{OUT} Fall Time	t_F			100	ns	See Test Circuits, Figure 1-2 Note 1
Power Requirements						
Operating Voltage	V_{DD}	2.7		5.5	V	
Operating Current	I_{DD}		375	550	μA	$V_{DD} = 5.0\text{V}$, D_{OUT} unloaded
Standby Current	I_{DDS}		0.5	5	μA	$CS = V_{DD} = 5.0\text{V}$

Note 1: This parameter is guaranteed by characterization and not 100% tested.

Note 2: Because the sample cap will eventually lose charge, effective clock rates below 10kHz can affect linearity performance, especially at elevated temperatures. See Section 6.2 for more information.

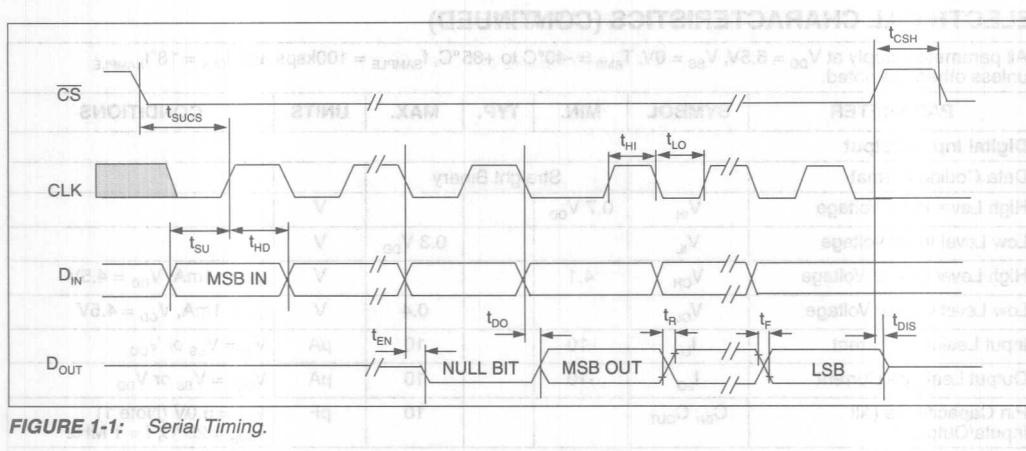


FIGURE 1-1: Serial Timing.

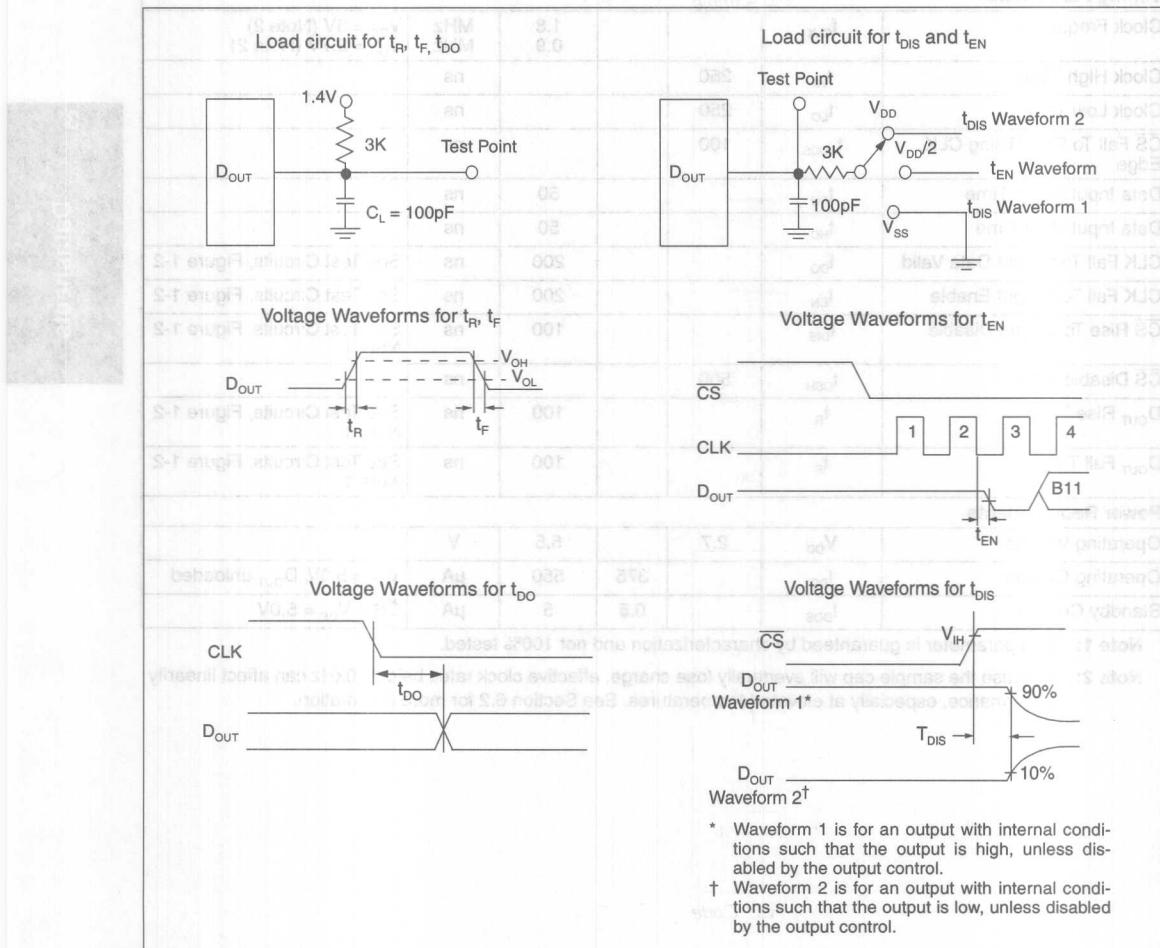


FIGURE 1-2: Test Circuits.

2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note: Unless otherwise indicated, $V_{DD} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{kspS}$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

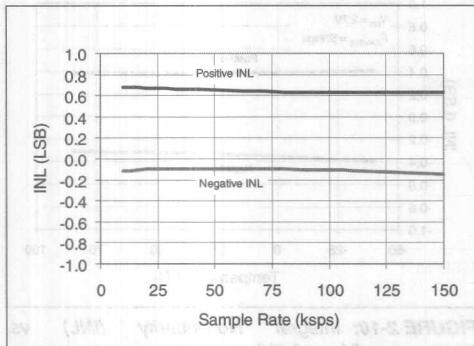


FIGURE 2-1: Integral Nonlinearity (INL) vs. Sample Rate.

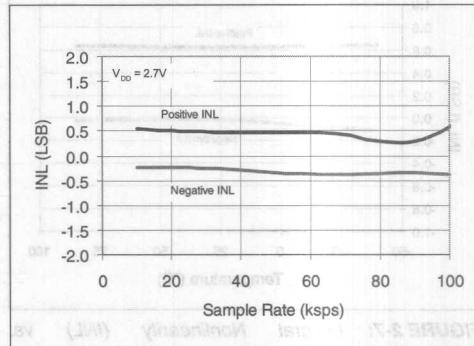


FIGURE 2-4: Integral Nonlinearity (INL) vs. Sample Rate ($V_{DD} = 2.7V$).

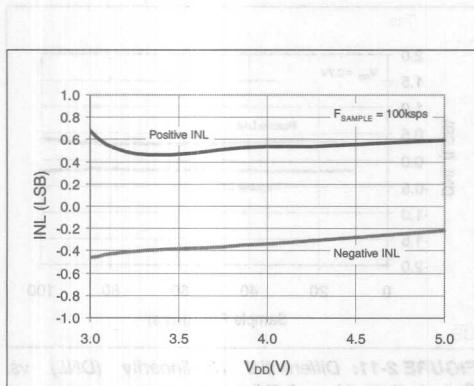


FIGURE 2-2: Integral Nonlinearity (INL) vs. V_{DD} .

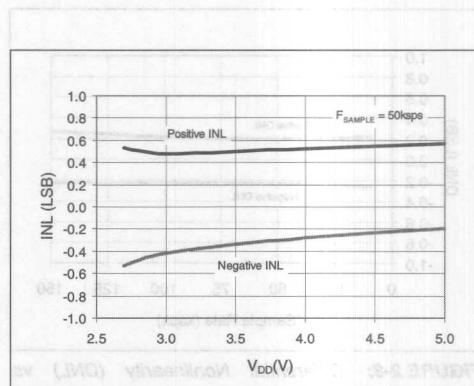


FIGURE 2-5: Integral Nonlinearity (INL) vs. V_{DD} .

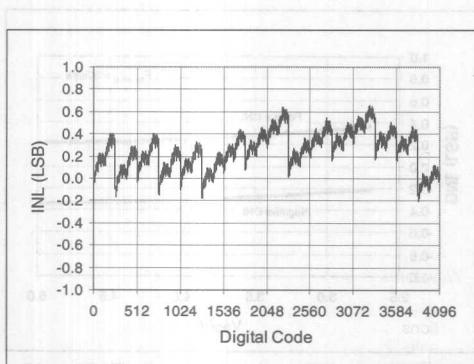


FIGURE 2-3: Integral Nonlinearity (INL) vs. Code (Representative Part).

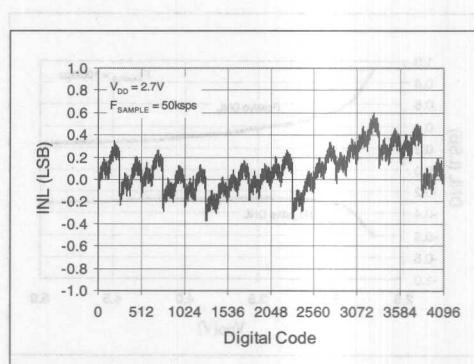


FIGURE 2-6: Integral Nonlinearity (INL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

MCP3202

Note: Unless otherwise indicated, $V_{DD} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 18 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

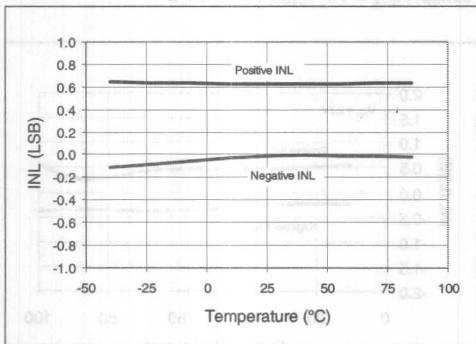


FIGURE 2-7: Integral Nonlinearity (INL) vs. Temperature.

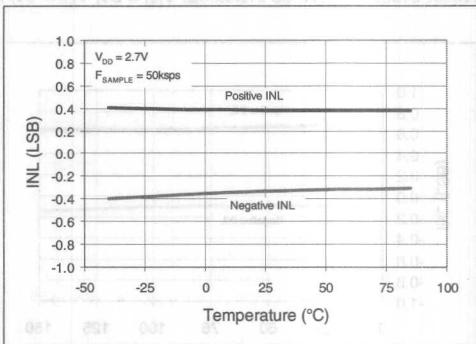


FIGURE 2-10: Integral Nonlinearity (INL) vs. Temperature ($V_{DD} = 2.7V$).

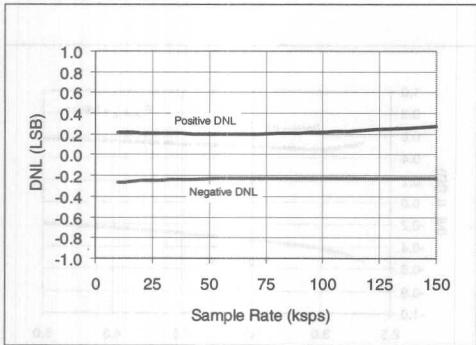


FIGURE 2-8: Differential Nonlinearity (DNL) vs. Sample Rate.

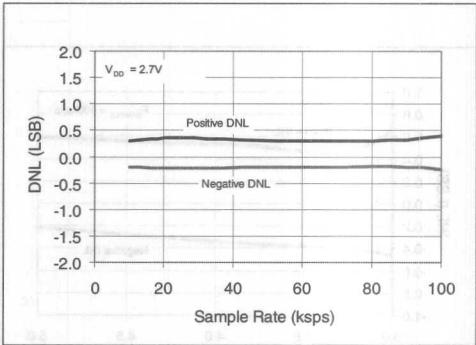


FIGURE 2-11: Differential Nonlinearity (DNL) vs. Sample Rate ($V_{DD} = 2.7V$).

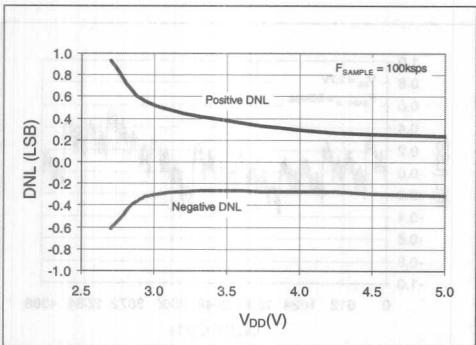


FIGURE 2-9: Differential Nonlinearity (DNL) vs. V_{DD} .

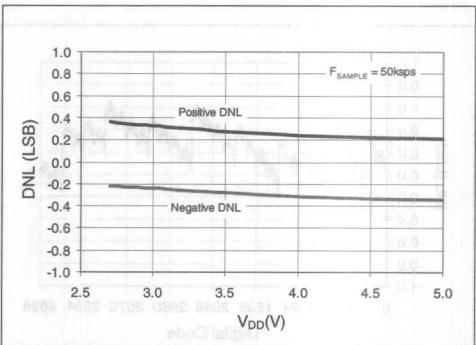


FIGURE 2-12: Differential Nonlinearity (DNL) vs. V_{DD} .

Note: Unless otherwise indicated, $V_{DD} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 18 \times f_{SAMPLE}$; $T_A = 25^\circ\text{C}$

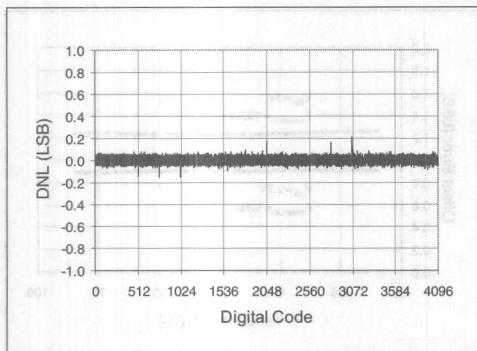


FIGURE 2-13: Differential Nonlinearity (DNL) vs. Code (Representative Part).

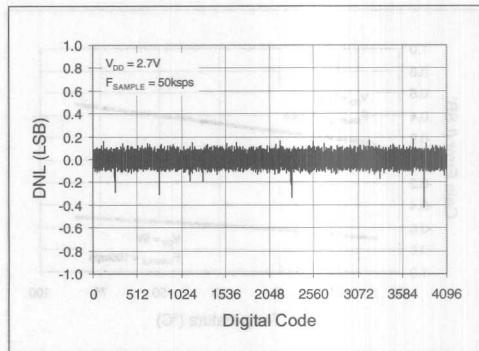


FIGURE 2-16: Differential Nonlinearity (DNL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

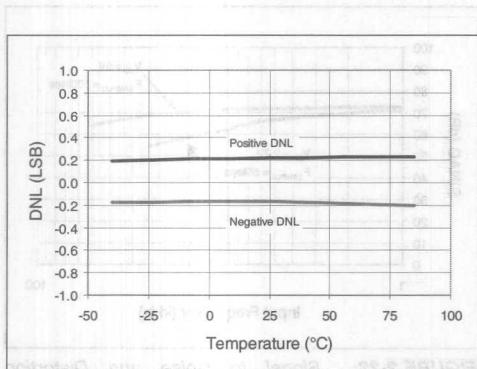


FIGURE 2-14: Differential Nonlinearity (DNL) vs. Temperature.

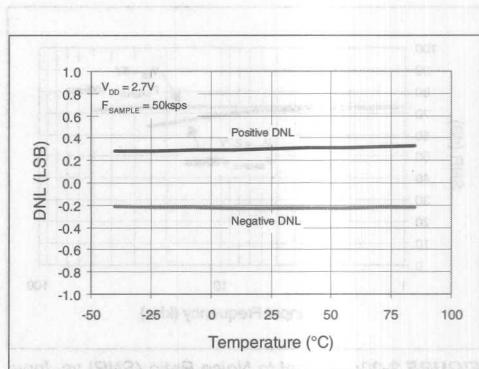


FIGURE 2-17: Differential Nonlinearity (DNL) vs. Temperature ($V_{DD} = 2.7V$).

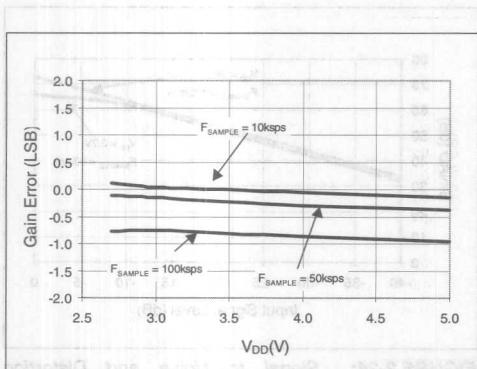


FIGURE 2-15: Gain Error vs. V_{DD} .

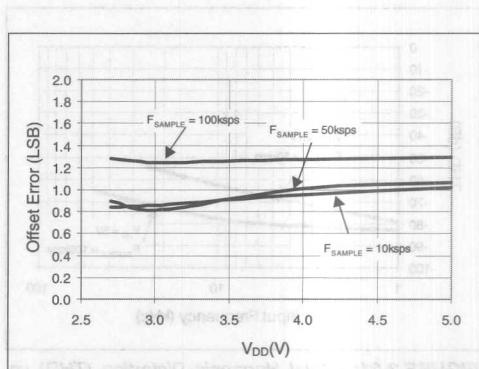


FIGURE 2-18: Offset Error vs. V_{DD} .

MCP3202

Note: Unless otherwise indicated, $V_{DD} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 18 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

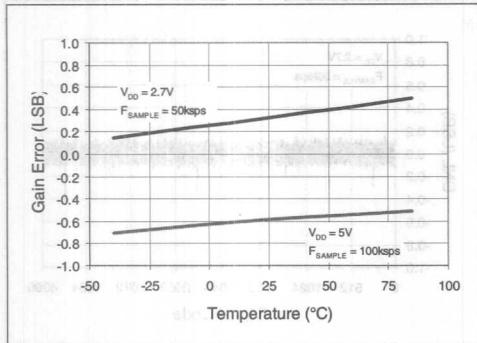


FIGURE 2-19: Gain Error vs. Temperature.

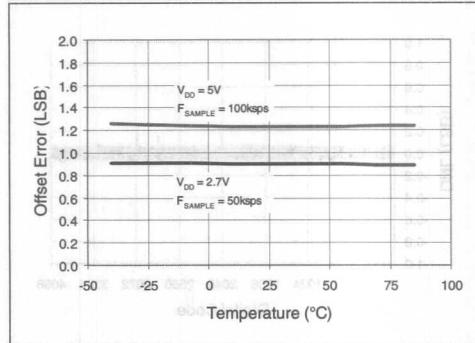


FIGURE 2-22: Offset Error vs. Temperature.

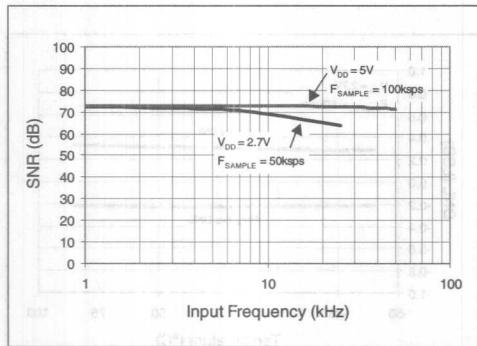


FIGURE 2-20: Signal to Noise Ratio (SNR) vs. Input Frequency.

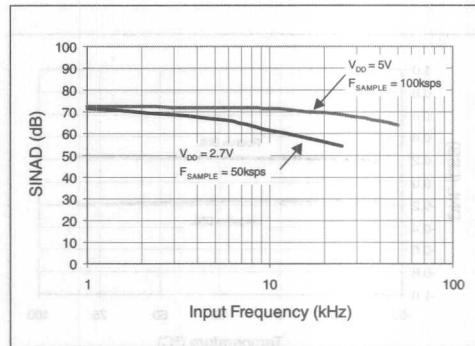


FIGURE 2-23: Signal to Noise and Distortion (SINAD) vs. Input Frequency.

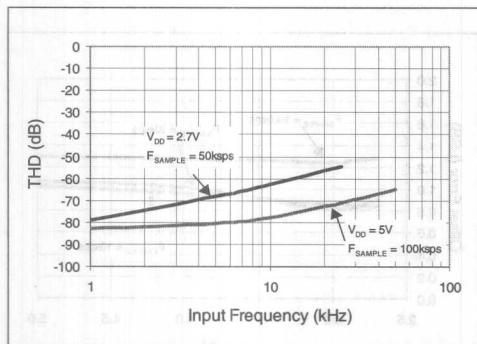


FIGURE 2-21: Total Harmonic Distortion (THD) vs. Input Frequency.

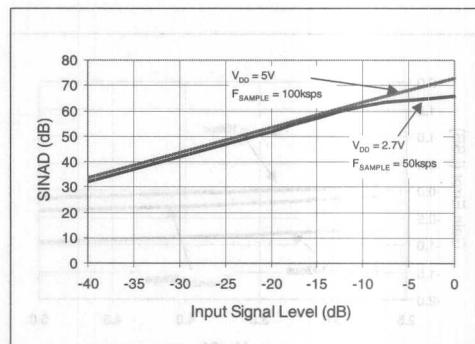


FIGURE 2-24: Signal to Noise and Distortion (SINAD) vs. Signal Level.

Note: Unless otherwise indicated, $V_{DD} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 18 * f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

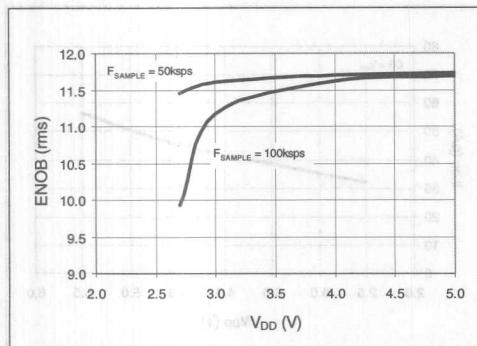


FIGURE 2-25: Effective number of bits (ENOB) vs. V_{DD} .

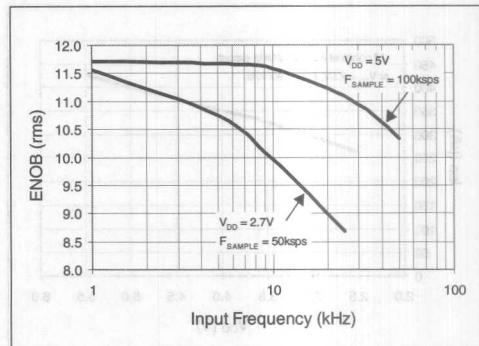


FIGURE 2-28: Effective Number of Bits (ENOB) vs. Input Frequency.

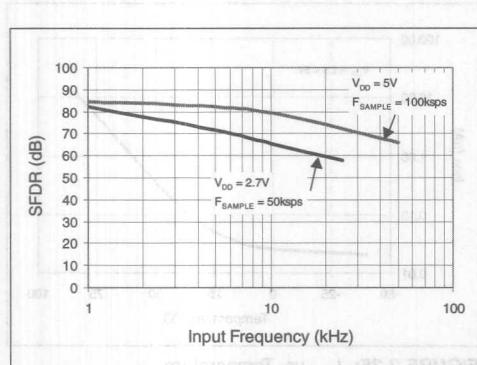


FIGURE 2-26: Spurious Free Dynamic Range (SFDR) vs. Input Frequency.

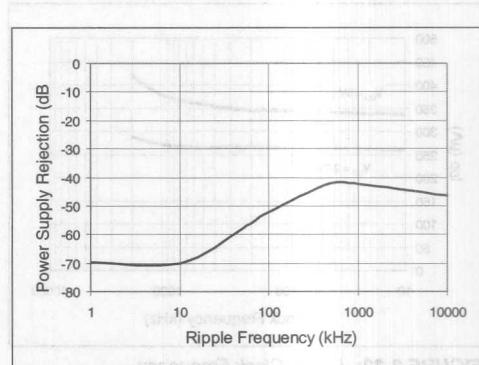


FIGURE 2-29: Power Supply Rejection (PSR) vs. Ripple Frequency.

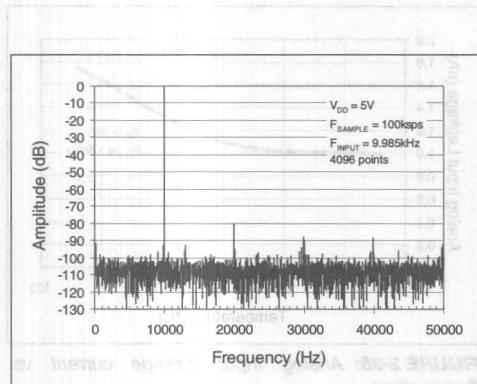


FIGURE 2-27: Frequency Spectrum of 10kHz input (Representative Part).

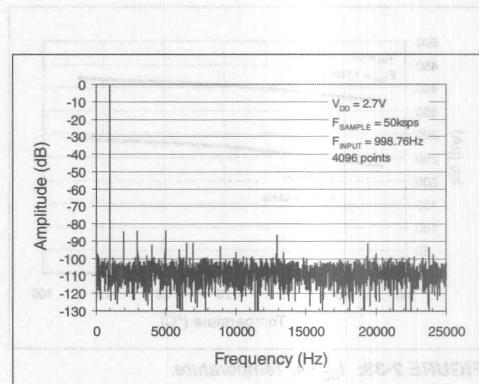


FIGURE 2-30: Frequency Spectrum of 1kHz input (Representative Part, $V_{DD} = 2.7V$).

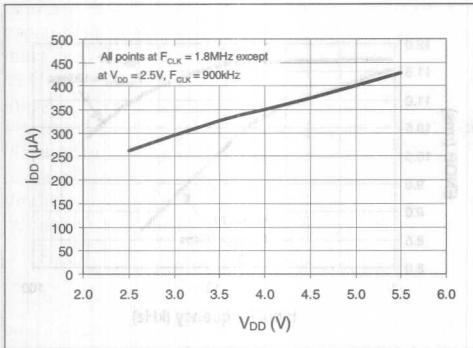


FIGURE 2-31: I_{DD} vs. V_{DD} .

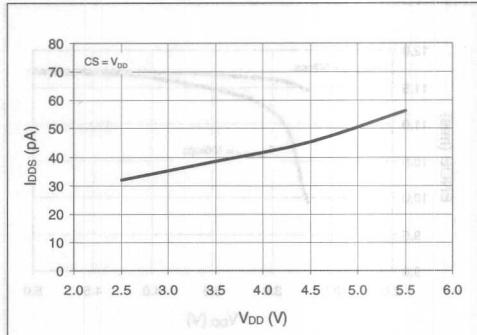


FIGURE 2-34: I_{DSS} vs. V_{DD} .

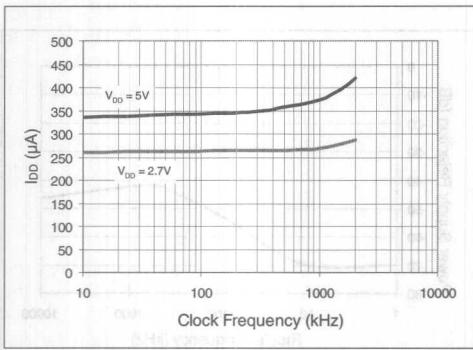


FIGURE 2-32: I_{DD} vs. Clock Frequency.

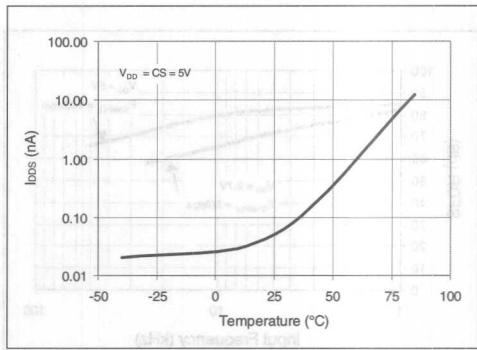


FIGURE 2-35: I_{DSS} vs. Temperature.

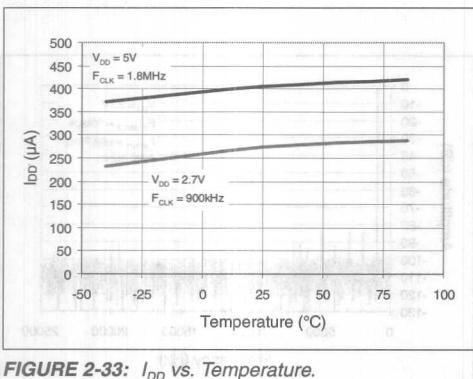


FIGURE 2-33: I_{DD} vs. Temperature.

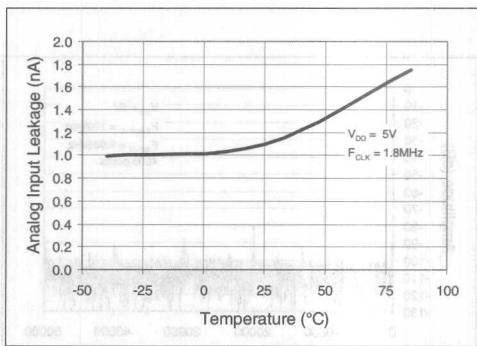


FIGURE 2-36: Analog Input leakage current vs. Temperature.

3.0 PIN DESCRIPTIONS

3.1 CH0/CH1

Analog inputs for channels 0 and 1 respectively. These channels can be programmed to be used as two independent channels in single ended-mode or as a single pseudo-differential input where one channel is IN+ and one channel is IN-. See Section 5.0 for information on programming the channel configuration.

3.2 CS/SHDN (Chip Select/Shutdown)

The CS/SHDN pin is used to initiate communication with the device when pulled low and will end a conversion and put the device in low power standby when pulled high. The CS/SHDN pin must be pulled high between conversions.

3.3 CLK (Serial Clock)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 6.2 for constraints on clock speed.

3.4 DIN (Serial Data Input)

The SPI port serial data input pin is used to clock in input channel configuration data.

3.5 DOUT (Serial Data Output)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

4.0 DEVICE OPERATION

The MCP3202 A/D Converter employs a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the second rising edge of the serial clock after the start bit has been received. Following this sample time, the input switch of the converter opens and the device uses the collected charge on the internal sample and hold capacitor to produce a serial 12-bit digital output code. Conversion rates of 100ksps are possible on the MCP3202. See Section 6.2 for information on minimum clock rates. Communication with the device is done using a 3-wire SPI-compatible interface.

4.1 Analog Inputs

The MCP3202 device offers the choice of using the analog input channels configured as two single-ended inputs or a single pseudo-differential input. Configuration is done as part of the serial command before each conversion begins. When used in the pseudo-differential mode, CH0 and CH1 are programmed as the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ input can range from IN- to V_{REF} ($V_{REF} + IN-$). The IN- input is limited to $\pm 100mV$ from the V_{SS} rail. The IN- input can be used to cancel small signal common-mode noise which is present on both the IN+ and IN- inputs.

For the A/D Converter to meet specification, the charge holding capacitor (C_{SAMPLE}) must be given enough time to acquire a 12-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

In this diagram, it is shown that the source impedance (R_s) adds to the internal sampling switch (R_{SS}) impedance, directly affecting the time that is required to charge the capacitor, C_{SAMPLE} . Consequently, larger source impedances increase the offset, gain, and integral linearity errors of the conversion.

Ideally, the impedance of the signal source should be near zero. This is achievable with an operational amplifier such as the MCP601 which has a closed loop output impedance of tens of ohms. The adverse affects of higher source impedances are shown in Figure 4-2.

When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h. If the voltage at IN+ is equal to or greater than [$V_{REF} + (IN-)] - 1$ LSB], then the output code will be FFFh. If the voltage level at IN- is more than 1 LSB below V_{SS} , then the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS} , then the FFFh code will not be seen unless the IN+ input level goes above V_{REF} level.

4.2 Digital Output Code

The digital output code produced by an A/D Converter is a function of the input signal and the reference voltage. For the MCP3202, V_{DD} is used as the reference voltage. As the V_{DD} level is reduced, the LSB size is reduced accordingly. The theoretical digital output code produced by the A/D Converter is shown below.

$$\text{Digital Output Code} = \frac{4096 * V_{IN}}{V_{DD}}$$

where:

V_{IN} = analog input voltage

V_{DD} = supply voltage

MCP3202

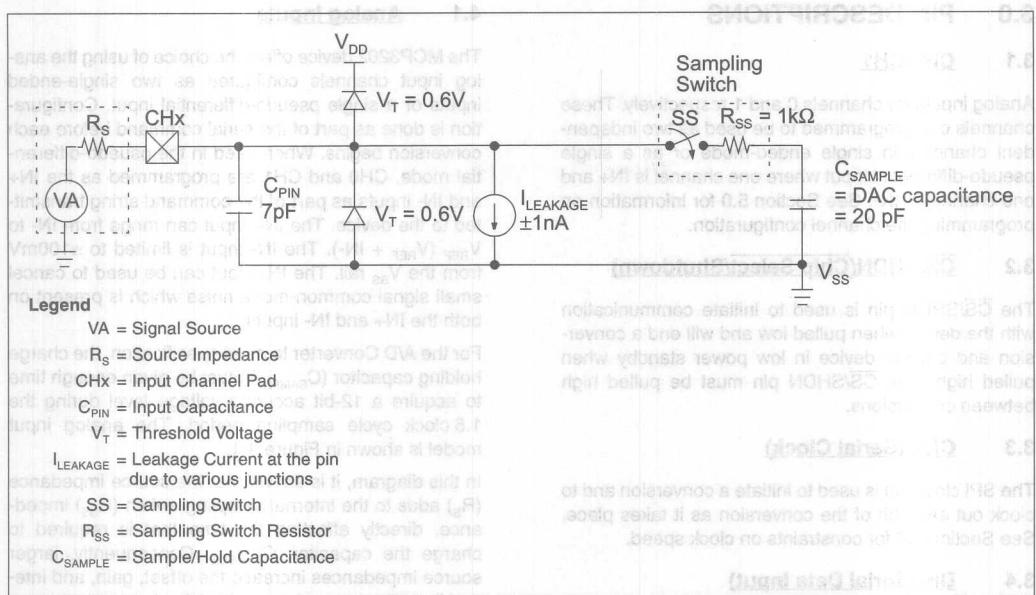


FIGURE 4-1: Analog Input Model.

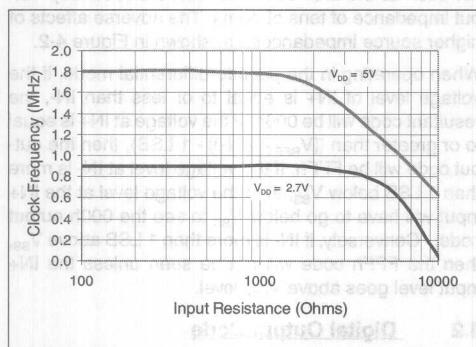


FIGURE 4-2: Maximum Clock Frequency vs. Input Resistance (R_s) to maintain less than a 0.1 LSB deviation in INL from nominal conditions.

$$\text{Clock Frequency} = \frac{V_{DD}}{4R_s + 1}$$

where

$$R_s = \text{Source Impedance}$$

$$V_{DD} = \text{Supply Voltage}$$

$$A_V = \text{Gain}$$

5.0 SERIAL COMMUNICATIONS

5.1 Overview

Communication with the MCP3202 is done using a standard SPI-compatible serial interface. Initiating communication with the device is done by bringing the CS line low. See Figure 5-1. If the device was powered up with the CS pin low, it must be brought high and back low to initiate communication. The first clock received with CS low and DIN high will constitute a start bit. The SGL/DIFF bit and the ODD/SIGN bit follow the start bit and are used to select the input channel configuration. The SGL/DIFF is used to select single ended or pseudo-differential mode. The ODD/SIGN bit selects which channel is used in single ended mode, and is used to determine polarity in pseudo-differential mode. Following the ODD/SIGN bit, the MSBF bit is transmitted to and is used to enable the LSB first format for the device. If the MSBF bit is low, then the data will come from the device in MSB first format and any further clocks with CS low will cause the device to output zeros. If the MSBF bit is high, then the device will output the converted word LSB first after the word has been transmitted in the MSB first format. See Figure 5-2. Table 5-1 shows the configuration bits for the MCP3202. The device will begin to sample the analog input on the second rising edge of the clock, after the start bit has been received. The sample period will end on the falling edge of the third clock following the start bit.

On the falling edge of the clock for the MSBF bit, the device will output a low null bit. The next sequential 12 clocks will output the result of the conversion with

MSB first as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 12 data bits have been transmitted and the device continues to receive clocks while the CS is held low, (and MSBF = 1), the device will output the conversion result LSB first as shown in Figure 5-2. If more clocks are provided to the device while CS is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring CS low and clock in leading zeros on the DIN line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.1 for more details on using the MCP3202 devices with hardware SPI ports.

	CONFIG BITS		CHANNEL SELECTION		GND
	SGL/DIFF	ODD/SIGN	0	1	
SINGLE ENDED MODE	1	0	+	-	
	1	1		+	-
PSEUDO-DIFFERENTIAL MODE	0	0	IN+	IN-	
	0	1	IN-	IN+	

TABLE 5-1: Configuration Bits for the MCP3202.

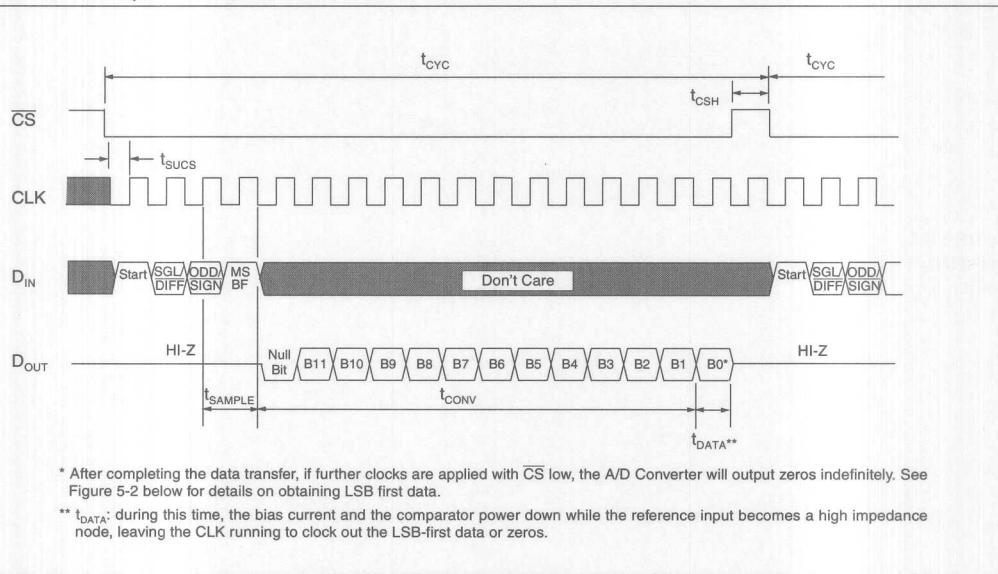


FIGURE 5-1: Communication with the MCP3202 using MSB first format only.

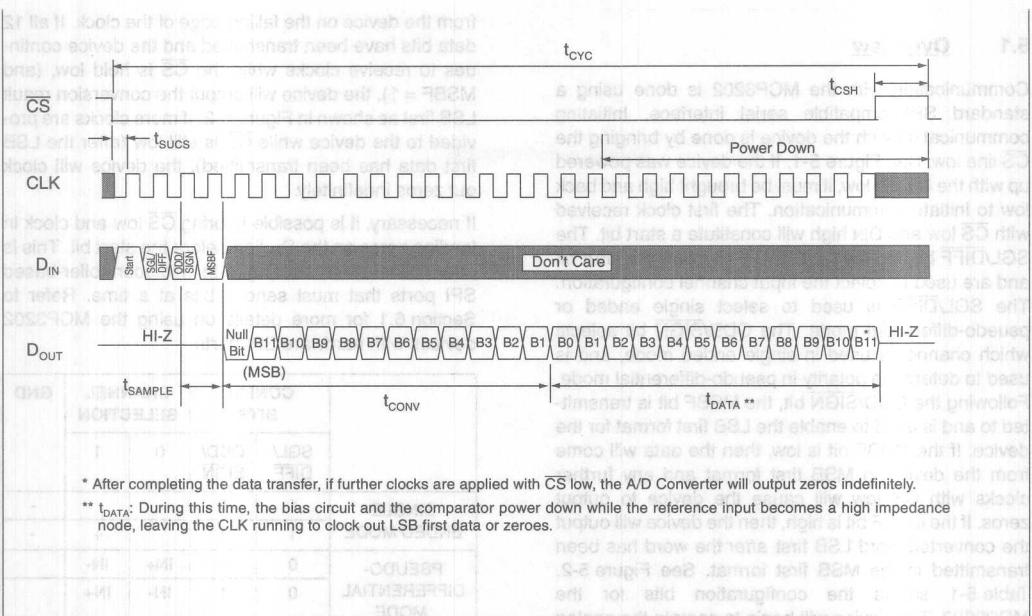
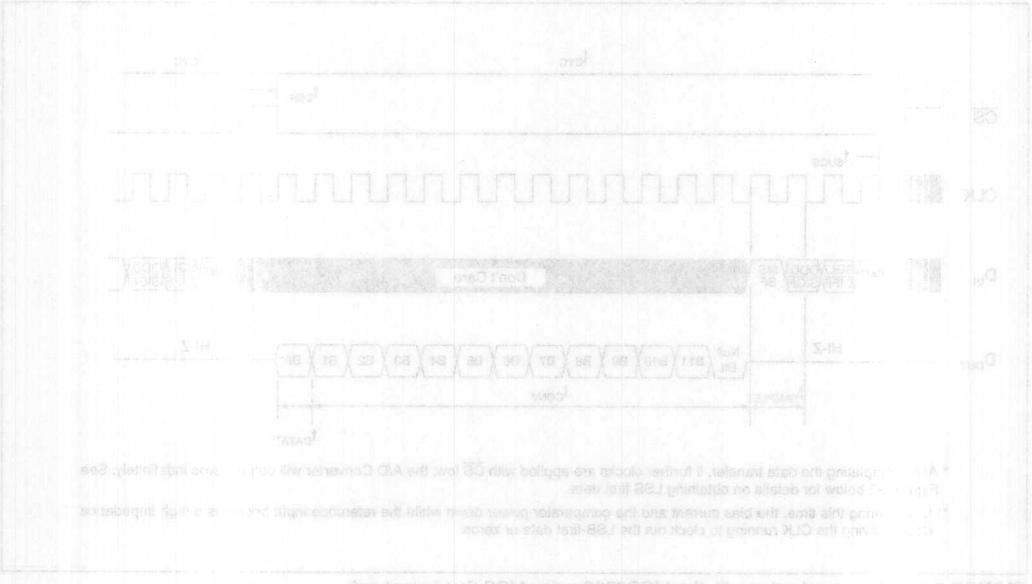


FIGURE 5-2: Communication with MCP3202 using LSB first format.



6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3202 with Microcontroller (MCU) SPI Ports

With most microcontroller SPI ports, it is required to send groups of eight bits. It is also required that the microcontroller SPI port be configured to clock out data on the falling edge of clock and latch data in on the rising edge. Depending on how communication routines are used, it is very possible that the number of clocks required for communication will not be a multiple of eight. Therefore, it may be necessary for the MCU to send more clocks than are actually required. This is usually done by sending 'leading zeros' before the start bit, which are ignored by the device. As an example, Figure 6-1 and Figure 6-2 show how the MCP3202 can be interfaced to a MCU with a hardware SPI port. Figure 6-1 depicts the operation shown in SPI Mode 0,0,

which requires that the SCLK from the MCU idles in the 'low' state, while Figure 6-2 shows the similar case of SPI Mode 1,1 where the clock idles in the 'high' state.

As shown in Figure 6-1, the first byte transmitted to the A/D Converter contains seven leading zeros before the start bit. Arranging the leading zeros this way produces the output 12 bits to fall in positions easily manipulated by the MCU. The MSB is clocked out of the A/D Converter on the falling edge of clock number 12. After the second eight clocks have been sent to the device, the MCU receive buffer will contain three unknown bits (the output is at high impedance until the null bit is clocked out), the null bit and the highest order four bits of the conversion. After the third byte has been sent to the device, the receive register will contain the lowest order eight bits of the conversion results. Easier manipulation of the converted data can be obtained by using this method.

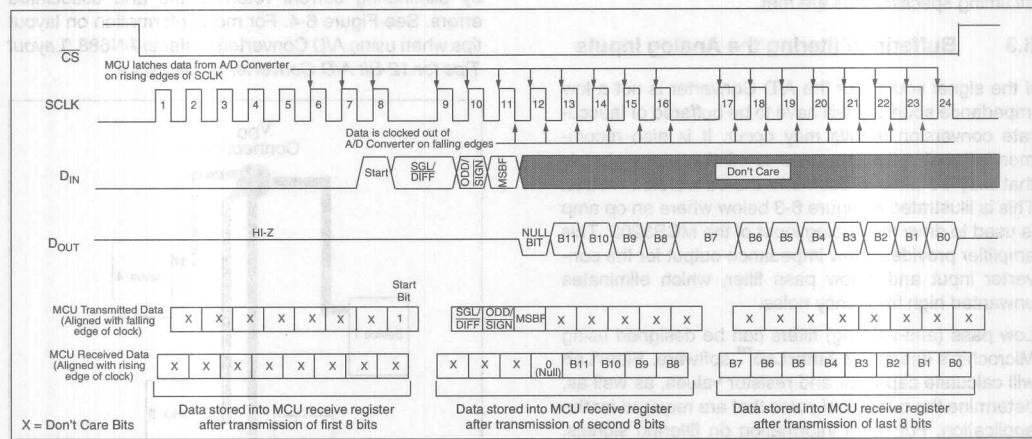


FIGURE 6-1: SPI Communication using 8-bit segments (Mode 0,0: SCLK idles low).

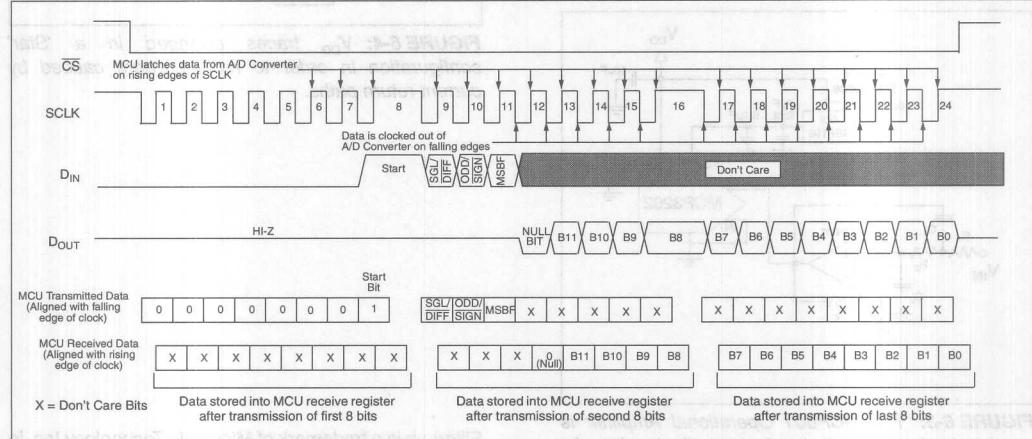


FIGURE 6-2: SPI Communication using 8-bit segments (Mode 1,1: SCLK idles high).

6.2 Maintaining Minimum Clock Speed

When the MCP3202 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample cap while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample capacitor for at least 1.2ms after the sample period has ended. This means that the time between the end of the sample period and the time that all 12 data bits have been clocked out must not exceed 1.2ms (effective clock frequency of 10kHz). Failure to meet this criteria may induce linearity errors into the conversion outside the rated specifications. It should be noted that during the entire conversion cycle, the A/D Converter does not require a constant clock speed or duty cycle, as long as all timing specifications are met.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the A/D Converter is not a low impedance source, it will have to be buffered or inaccurate conversion results may occur. It is also recommended that a filter be used to eliminate any signals that may be aliased back into the conversion results. This is illustrated in Figure 6-3 below where an op amp is used to drive the analog input of the MCP3202. This amplifier provides a low impedance output for the converter input and a low pass filter, which eliminates unwanted high frequency noise.

Low pass (anti-aliasing) filters can be designed using Microchip's interactive **FilterLab™** software. **FilterLab** will calculate capacitor and resistor values, as well as, determine the number of poles that are required for the application. For more information on filtering signals, see the application note AN699 "Anti-Aliasing Analog Filters for Data Acquisition Systems."

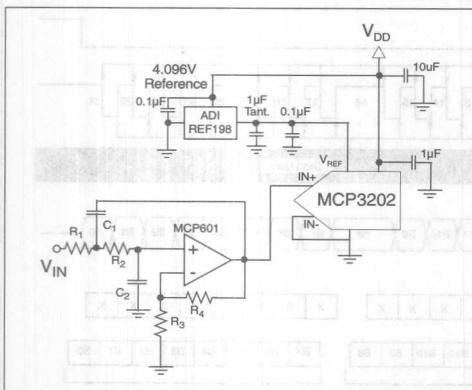


FIGURE 6-3: The MCP601 Operational Amplifier is used to implement a 2nd order anti-aliasing filter for the signal being converted by the MCP3202.

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of $1\mu F$ is recommended.

Digital and analog traces should be separated as much as possible on the board and no traces should run underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating current return paths and associated errors. See Figure 6-4. For more information on layout tips when using A/D Converters, refer to AN688 "Layout Tips for 12-Bit A/D Converter Applications".

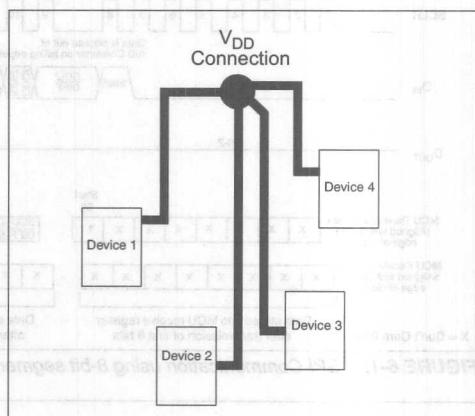


FIGURE 6-4: V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.

FilterLab is a trademark of Microchip Technology Inc. in the U.S.A and other countries. All rights reserved.

MCP3202 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

MCP3202 - G T /P		
	Package:	P = PDIP (8 lead) SN = SOIC (150 mil Body), 8 lead ST = TSSOP, 8 lead (C Grade only)
	Temperature Range:	I = -40°C to +85°C
	Performance Grade:	B = ±1 LSB INL (TSSOP not available in this grade) C = ±2 LSB INL
	Device:	MCP3202 = 12-Bit Serial A/D Converter MCP3202T = 12-Bit Serial A/D Converter on tape and reel (SOIC and TSSOP packages only)

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277. After September 1, 1999 (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

3

Datasheets

2.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI™ Serial Interface

FEATURES

- 12-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL (MCP3204/3208-B)
- ± 2 LSB max INL (MCP3204/3208-C)
- 4 (MCP3204) or 8 (MCP3208) input channels
- Analog inputs programmable as single-ended or pseudo differential pairs
- On-chip sample and hold
- SPI™ serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 100ksps max. sampling rate at $V_{DD} = 5V$
- 50ksps max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
 - 500 nA typical standby current, 2 μ A max.
 - 400 μ A max. active current at 5V
- Industrial temp range: -40°C to +85°C
- Available in PDIP, SOIC and TSSOP packages

APPLICATIONS

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

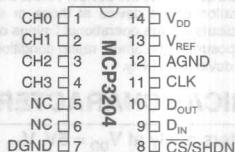
DESCRIPTION

The Microchip Technology Inc. MCP3204/3208 devices are successive approximation 12-bit Analog-to-Digital (A/D) Converters with on-board sample and hold circuitry. The MCP3204 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3208 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) is specified at ± 1 LSB, and Integral Nonlinearity (INL) is offered in ± 1 LSB (MCP3204/3208-B) and ± 2 LSB (MCP3204/3208-C) versions. Communication with the devices is done using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 100ksps. The MCP3204/3208 devices operate over a broad voltage range (2.7V - 5.5V). Low current design permits operation with typical standby and active currents of only 500nA and 320 μ A, respectively. The MCP3204 is offered in 14-pin PDIP, 150mil SOIC and TSSOP packages, and the MCP3208 is offered in 16-pin PDIP and SOIC packages.

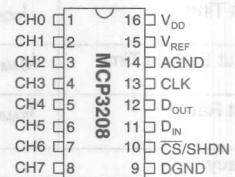
SPI is a trademark of Motorola Inc.

PACKAGE TYPES

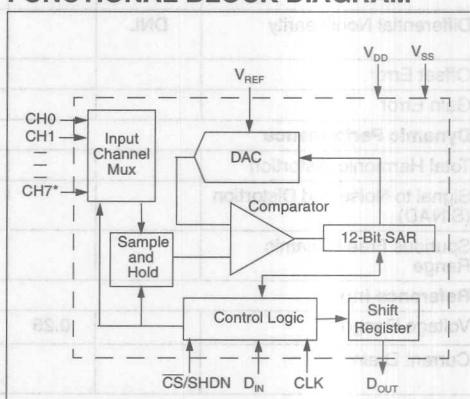
PDIP, SOIC, TSSOP



PDIP, SOIC



FUNCTIONAL BLOCK DIAGRAM



*Note: Channels 5-7 available on MCP3208 Only

MCP3204/3208

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

V_{DD} 7.0V
 All inputs and outputs w.r.t. V_{SS} -0.6V to V_{DD} +0.6V
 Storage temperature -65°C to +150°C
 Ambient temp. with power applied -65°C to +125°C
 Soldering temperature of leads (10 seconds) .. +300°C
 ESD protection on all pins > 4kV

*Notice: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

PIN FUNCTION TABLE

NAME	FUNCTION
V_{DD}	+2.7V to 5.5V Power Supply
DGND	Digital Ground
AGND	Analog Ground
CH0-CH7	Analog Inputs
CLK	Serial Clock
D_{IN}	Serial Data In
D_{OUT}	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input
V_{REF}	Reference Voltage Input

ELECTRICAL CHARACTERISTICS

All parameters apply at $V_{DD} = 5V$, $V_{SS} = 0V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 100\text{ksps}$ and $f_{CLK} = 20f_{SAMPLE}$, unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Conversion Rate						
Conversion Time	t_{CONV}			12	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}			100 50	ksps ksps	$V_{DD} = V_{REF} = 5V$ $V_{DD} = V_{REF} = 2.7V$
DC Accuracy						
Resolution			12		bits	
Integral Nonlinearity	INL	± 0.75 ± 1	± 1 ± 2		LSB	MCP3204/3208-B MCP3204/3208-C
Differential Nonlinearity	DNL		± 0.5	± 1	LSB	No missing codes over temperature
Offset Error			± 1.25	± 3	LSB	
Gain Error			± 1.25	± 5	LSB	
Dynamic Performance						
Total Harmonic Distortion		-82			dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Signal to Noise and Distortion (SINAD)		72			dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Spurious Free Dynamic Range		86			dB	$V_{IN} = 0.1V$ to $4.9V$ @ 1kHz
Reference Input						
Voltage Range		0.25		V_{DD}	V	Note 2
Current Drain		100 0.001		150 3	μA	$\overline{CS} = V_{DD} = 5V$
Analog Inputs						
Input Voltage Range for CH0-CH7 in Single-Ended Mode		V_{SS}		V_{REF}	V	
Input Voltage Range for IN+ In pseudo-differential Mode		IN-		$V_{REF}+IN-$		
Input Voltage Range for IN- In pseudo-differential Mode		$V_{SS}-100$		$V_{SS}+100$	mV	
Leakage Current			0.001	± 1	μA	

ELECTRICAL CHARACTERISTICS (CONTINUED)

All parameters apply at $V_{DD} = 5V$, $V_{SS} = 0V$, $V_{REF} = 5V$, $T_{AMB} = -40^{\circ}C$ to $+85^{\circ}C$, $f_{SAMPLE} = 100\text{ksps}$ and $f_{CLK} = 20^{\ast}f_{SAMPLE}$, unless otherwise noted.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Analog Inputs (Continued)						
Switch Resistance			1K		Ω	See Figure 4-1
Sample Capacitor			20		pF	See Figure 4-1
Digital Input/Output						
Data Coding Format			Straight Binary			
High Level Input Voltage	V_{IH}	0.7 V_{DD}			V	
Low Level Input Voltage	V_{IL}		0.3 V_{DD}		V	
High Level Output Voltage	V_{OH}	4.1			V	$I_{OH} = -1\text{mA}$, $V_{DD} = 4.5V$
Low Level Output Voltage	V_{OL}		0.4		V	$I_{OL} = 1\text{mA}$, $V_{DD} = 4.5V$
Input Leakage Current	I_{LI}	-10		10	μA	$V_{IN} = V_{SS}$ or V_{DD}
Output Leakage Current	I_{LO}	-10		10	μA	$V_{OUT} = V_{SS}$ or V_{DD}
Pin Capacitance (All Inputs/Outputs)	C_{IN}, C_{OUT}			10	pF	$V_{DD} = 5.0V$ (Note 1) $T_{AMB} = 25^{\circ}\text{C}$, $f = 1\text{ MHz}$
Timing Parameters						
Clock Frequency	f_{CLK}			2.0 1.0	MHz MHz	$V_{DD} = 5V$ (Note 3) $V_{DD} = 2.7V$ (Note 3)
Clock High Time	t_{HI}	250			ns	
Clock Low Time	t_{LO}	250			ns	
CS Fall To First Rising CLK Edge	$t_{SU CS}$	100			ns	
Data Input Setup Time	t_{SU}			50	ns	
Data Input Hold Time	t_{HD}			50	ns	
CLK Fall To Output Data Valid	t_{DO}			200	ns	See Test Circuits, Figure 1-2
CLK Fall To Output Enable	t_{EN}			200	ns	See Test Circuits, Figure 1-2
CS Rise To Output Disable	t_{DIS}			100	ns	See Test Circuits, Figure 1-2
CS Disable Time	t_{CSH}	500			ns	
D_{OUT} Rise Time	t_R			100	ns	See Test Circuits, Figure 1-2 (Note 1)
D_{OUT} Fall Time	t_F			100	ns	See Test Circuits, Figure 1-2 (Note 1)
Power Requirements						
Operating Voltage	V_{DD}	2.7		5.5	V	
Operating Current	I_{DD}		320 225	400	μA	$V_{DD} = V_{REF} = 5V$, D_{OUT} unloaded $V_{DD} = V_{REF} = 2.7V$, D_{OUT} unloaded
Standby Current	I_{DDS}		0.5	2	μA	$\bar{CS} = V_{DD} = 5.0V$

Note 1: This parameter is guaranteed by characterization and not 100% tested.

Note 2: See graphs that relate linearity performance to V_{REF} levels.

Note 3: Because the sample cap will eventually lose charge, effective clock rates below 10kHz can affect linearity performance, especially at elevated temperatures. See Section 6.2 for more information.

MCP3204/3208

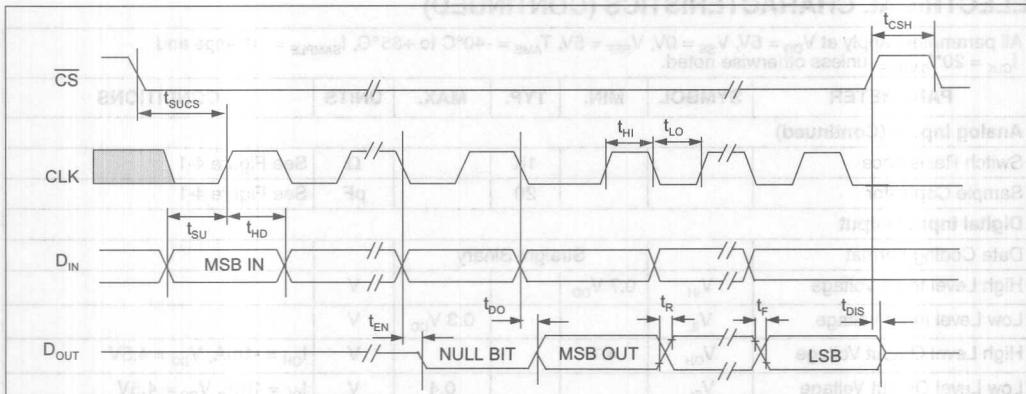


FIGURE 1-1: Serial Interface Timing.

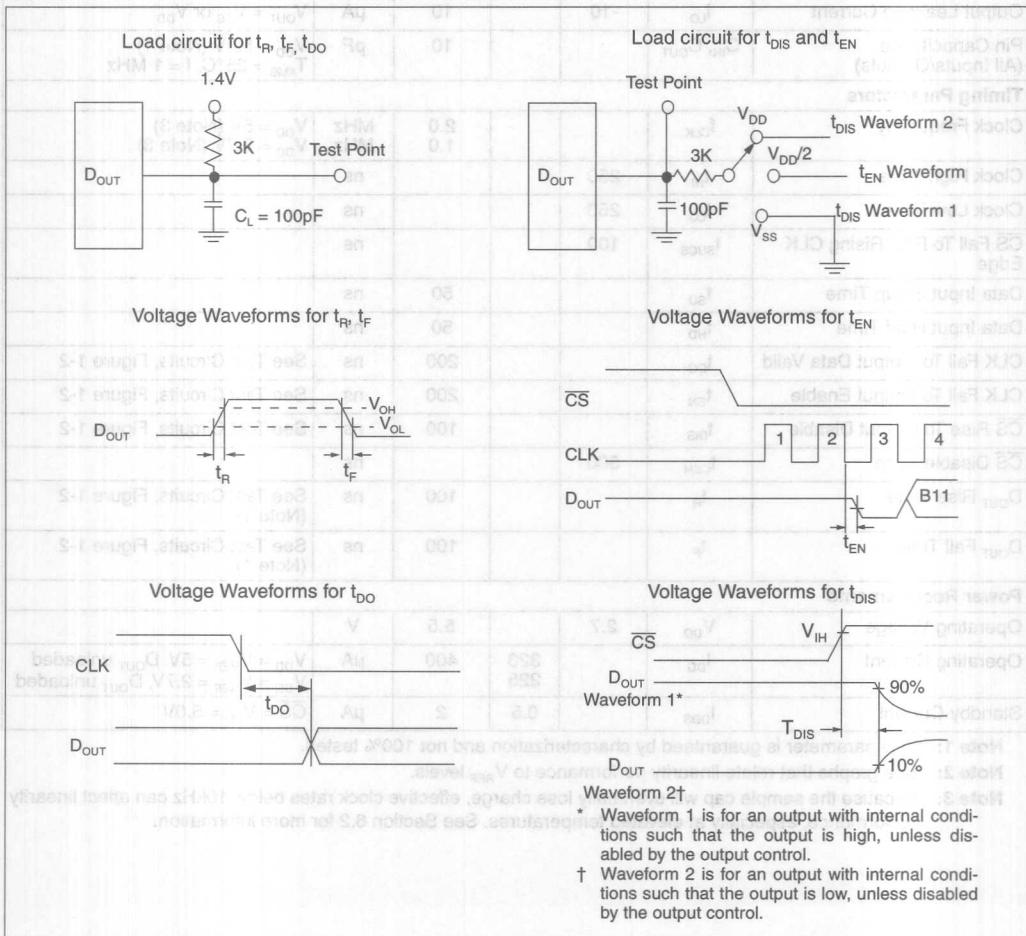


FIGURE 1-2: Test Circuits.

2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 20 \times f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

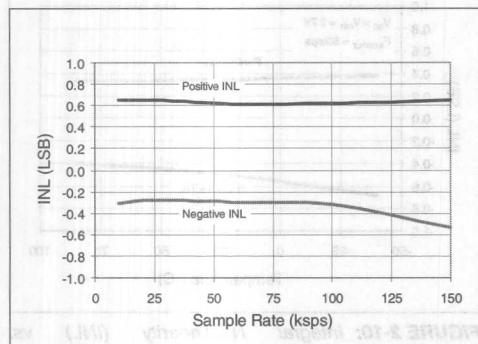


FIGURE 2-1: Integral Nonlinearity (INL) vs. Sample Rate.

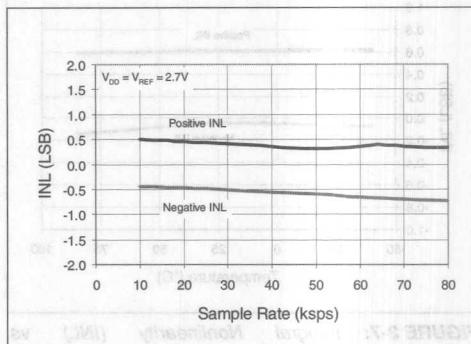


FIGURE 2-4: Integral Nonlinearity (INL) vs. Sample Rate ($V_{DD} = 2.7V$).

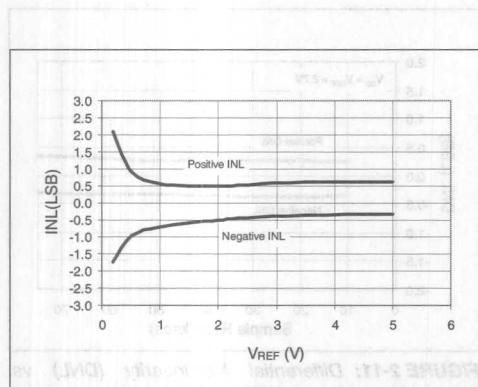


FIGURE 2-2: Integral Nonlinearity (INL) vs. V_{REF} .

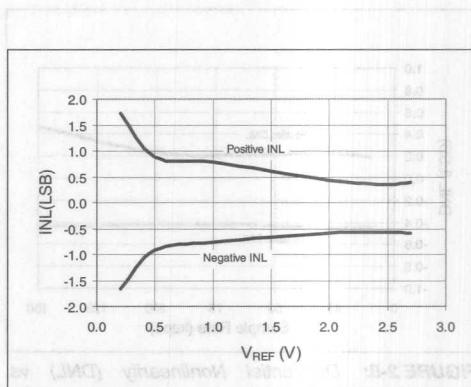


FIGURE 2-5: Integral Nonlinearity (INL) vs. V_{REF} ($V_{DD} = 2.7V$).

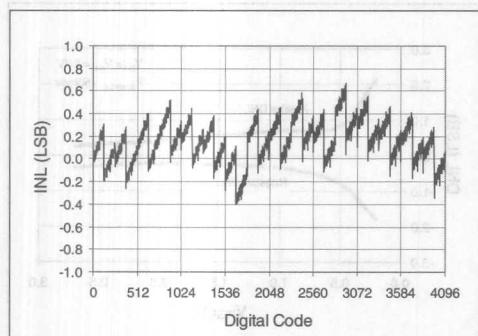


FIGURE 2-3: Integral Nonlinearity (INL) vs. Code (Representative Part).

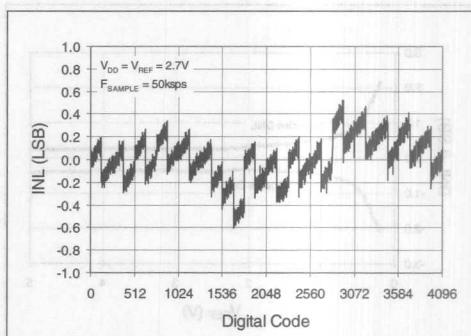


FIGURE 2-6: Integral Nonlinearity (INL) vs. Code ($V_{DD} = 2.7V$).

MCP3204/3208

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 20^* f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

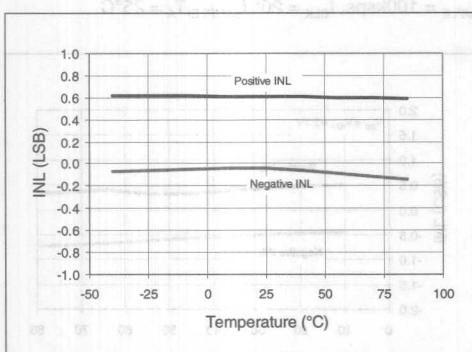


FIGURE 2-7: Integral Nonlinearity (INL) vs. Temperature.

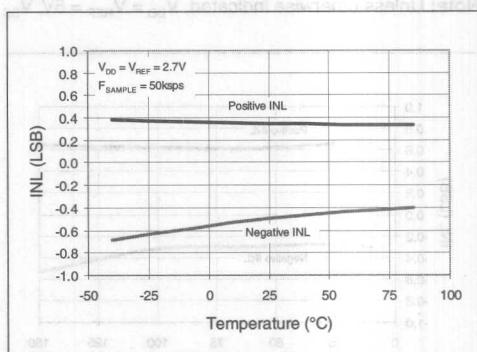


FIGURE 2-10: Integral Nonlinearity (INL) vs. Temperature ($V_{DD} = 2.7V$).

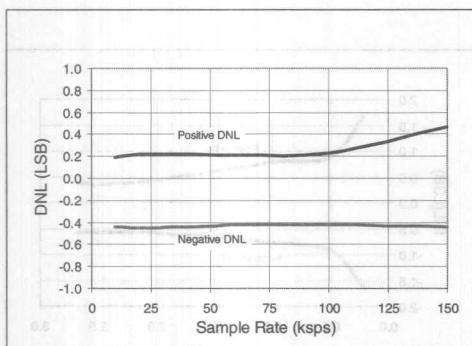


FIGURE 2-8: Differential Nonlinearity (DNL) vs. Sample Rate.

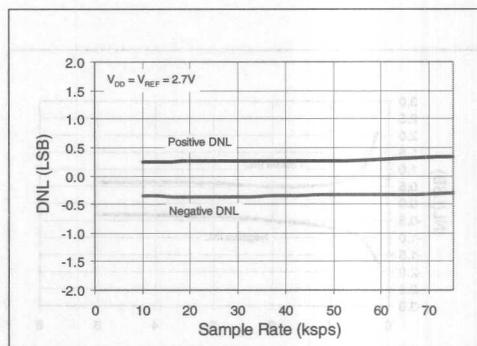


FIGURE 2-11: Differential Nonlinearity (DNL) vs. Sample Rate ($V_{DD} = 2.7V$).

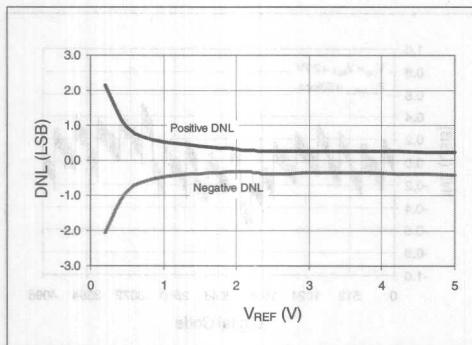


FIGURE 2-9: Differential Nonlinearity (DNL) vs. V_{REF} .

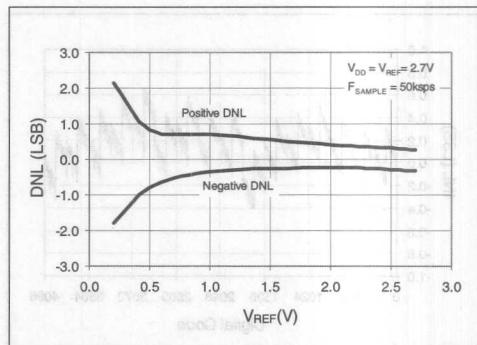


FIGURE 2-12: Differential Nonlinearity (DNL) vs. V_{REF} ($V_{DD} = 2.7V$).

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 20 * f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

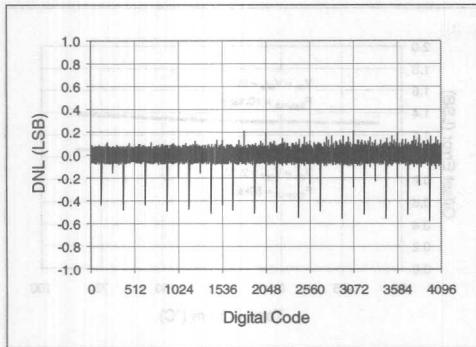


FIGURE 2-13: Differential Nonlinearity (DNL) vs. Code (Representative Part).

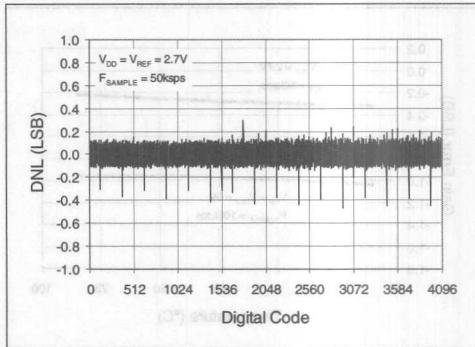


FIGURE 2-16: Differential Nonlinearity (DNL) vs. Code (Representative Part, $V_{DD} = 2.7V$).

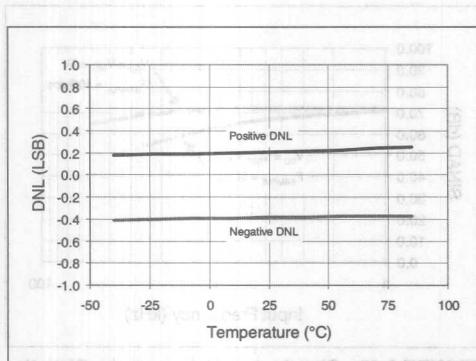


FIGURE 2-14: Differential Nonlinearity (DNL) vs. Temperature.

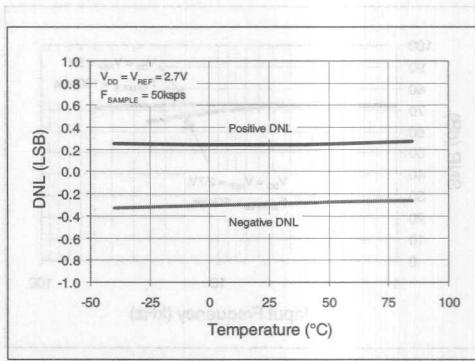


FIGURE 2-17: Differential Nonlinearity (DNL) vs. Temperature ($V_{DD} = 2.7V$).

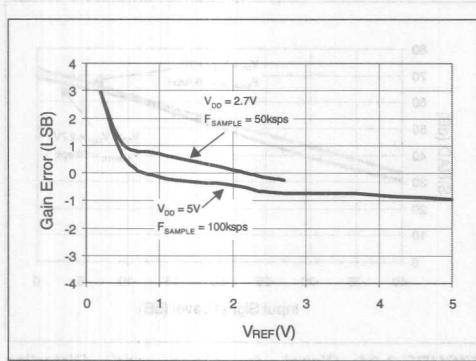


FIGURE 2-15: Gain Error vs. V_{REF} .

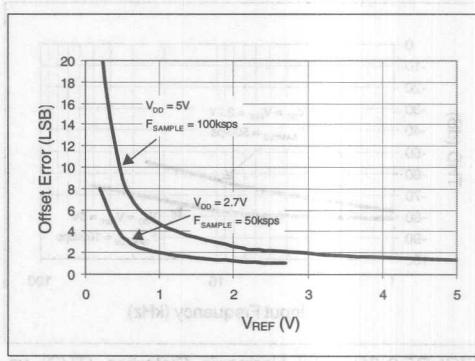


FIGURE 2-18: Offset Error vs. V_{REF} .

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 20 * f_{SAMPLE}$, $T_A = 25^\circ\text{C}$

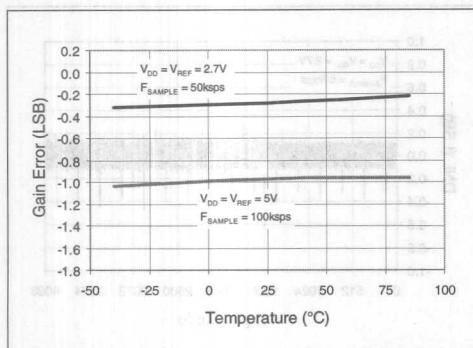


FIGURE 2-19: Gain Error vs. Temperature.

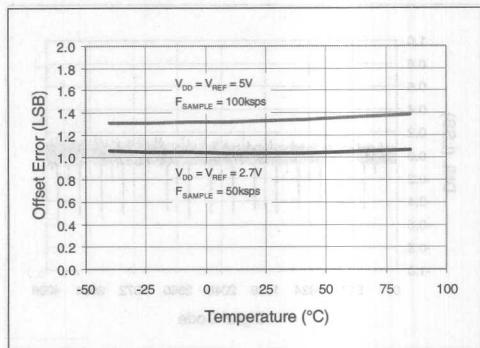


FIGURE 2-22: Offset Error vs. Temperature.

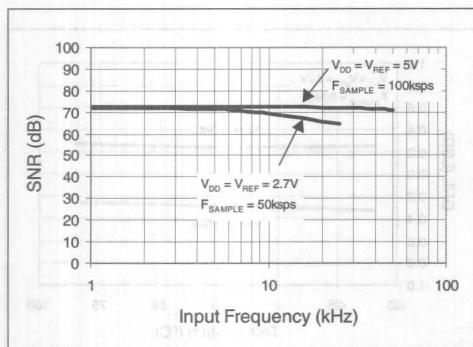


FIGURE 2-20: Signal to Noise (SNR) vs. Input Frequency.

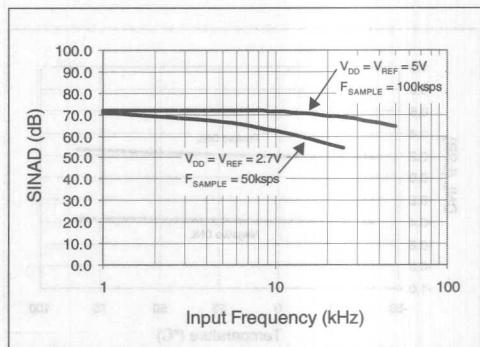


FIGURE 2-23: Signal to Noise and Distortion (SINAD) vs. Input Frequency.

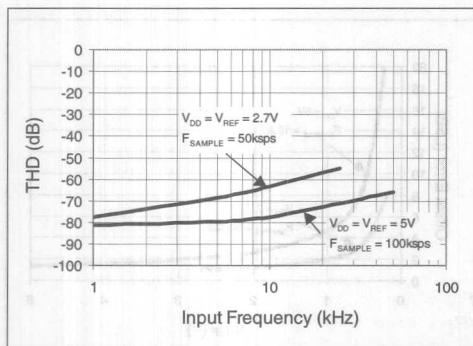


FIGURE 2-21: Total Harmonic Distortion (THD) vs. Input Frequency.

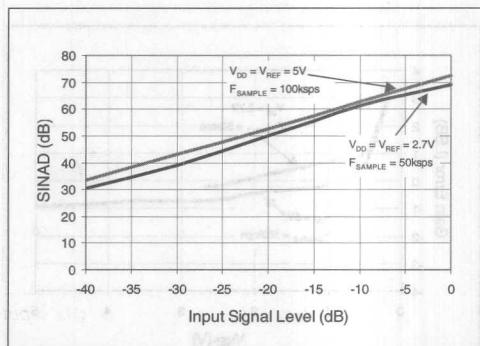


FIGURE 2-24: Signal to Noise and Distortion (SINAD) vs. Input Signal Level.

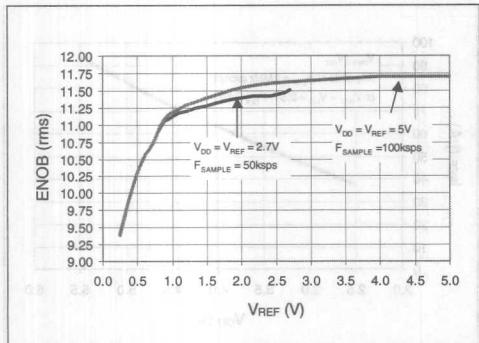


FIGURE 2-25: Effective Number of Bits (ENOB) vs. V_{REF} .

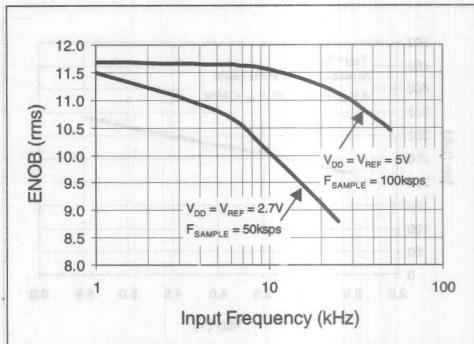


FIGURE 2-28: Effective Number of Bits (ENOB) vs. Input Frequency.

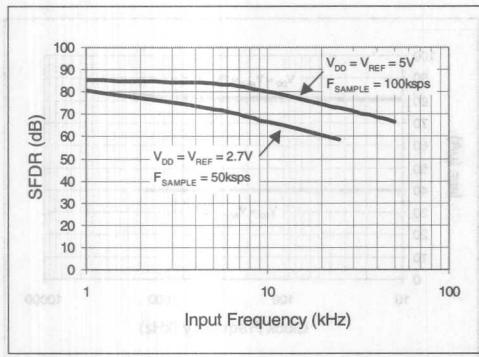


FIGURE 2-26: Spurious Free Dynamic Range (SFDR) vs. Input Frequency.

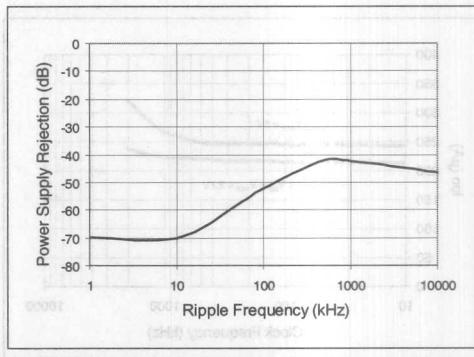


FIGURE 2-29: Power Supply Rejection (PSR) vs. Ripple Frequency.

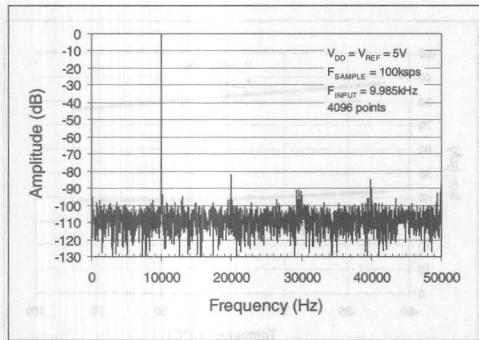


FIGURE 2-27: Frequency Spectrum of 10kHz input (Representative Part).

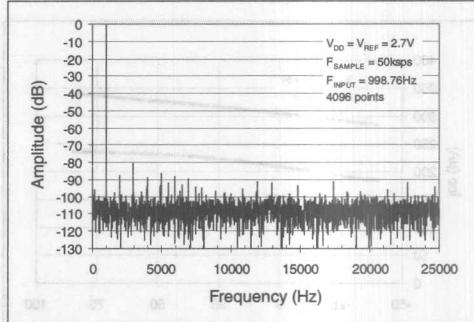
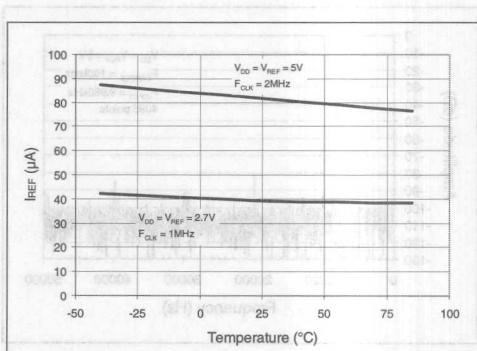
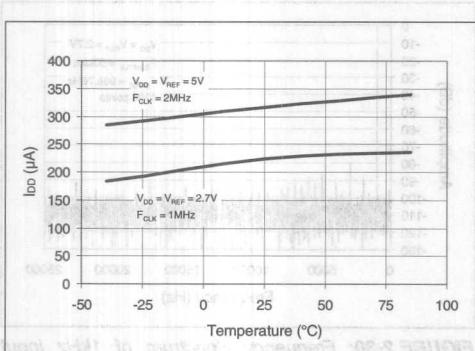
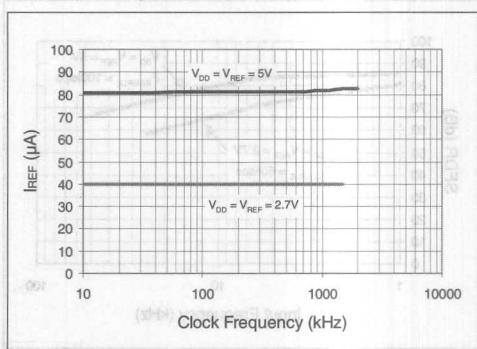
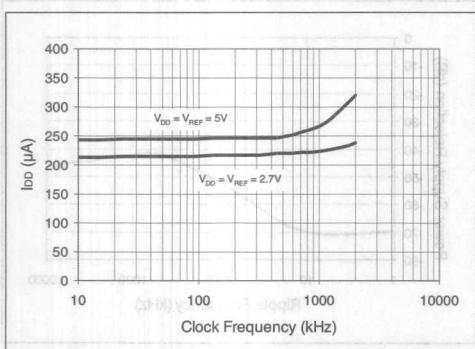
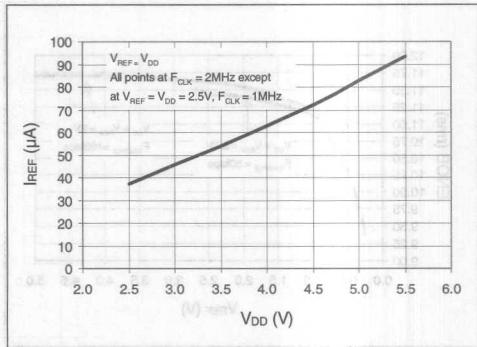
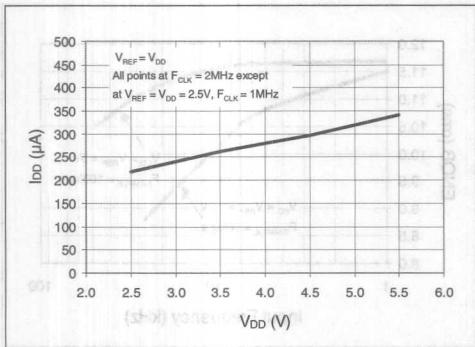


FIGURE 2-30: Frequency Spectrum of 1kHz input (Representative Part, $V_{DD} = 2.7V$).

MCP3204/3208

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 20 * f_{SAMPLE}$, $T_A = 25^\circ\text{C}$



Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $V_{SS} = 0V$, $f_{SAMPLE} = 100\text{ksps}$, $f_{CLK} = 20^*$ f_{SAMPLE} , $T_A = 25^\circ C$

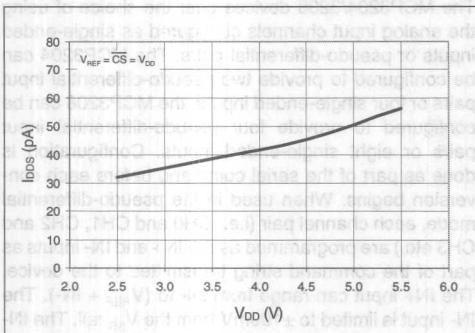


FIGURE 2-37: I_{DDS} vs. V_{DD} .

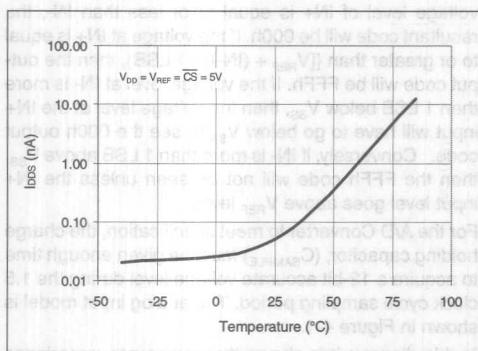


FIGURE 2-38: I_{DDS} vs. Temperature.

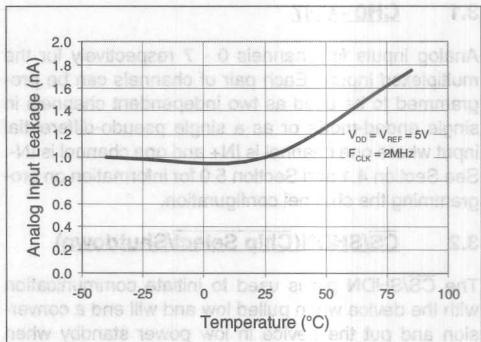


FIGURE 2-39: Analog Input Leakage Current vs. Temperature.

3

Datasheets

MCP3204/3208

3.0 PIN DESCRIPTIONS

3.1 CH0 - CH7

Analog inputs for channels 0 - 7 respectively for the multiplexed inputs. Each pair of channels can be programmed to be used as two independent channels in single ended-mode or as a single pseudo-differential input where one channel is IN+ and one channel is IN-. See Section 4.1 and Section 5.0 for information on programming the channel configuration.

3.2 CS/SHDN(Chip Select/Shutdown)

The CS/SHDN pin is used to initiate communication with the device when pulled low and will end a conversion and put the device in low power standby when pulled high. The CS/SHDN pin must be pulled high between conversions.

3.3 CLK (Serial Clock)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 6.2 for constraints on clock speed.

3.4 DIN (Serial Data Input)

The SPI port serial data input pin is used to load channel configuration data into the device.

3.5 DOUT (Serial Data output)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

3.6 AGND

Analog ground connection to internal analog circuitry.

3.7 DGND

Digital ground connection to internal digital circuitry.

4.0 DEVICE OPERATION

The MCP3204/3208 A/D Converters employ a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the fourth rising edge of the serial clock after the start bit has been received. Following this sample time, the device uses the collected charge on the internal sample and hold capacitor to produce a serial 12-bit digital output code. Conversion rates of 100ksps are possible on the MCP3204/3208. See Section 6.2 for information on minimum clock rates. Communication with the device is done using a 4-wire SPI-compatible interface.

4.1 V_I = Analog Inputs

The MCP3204/3208 devices offer the choice of using the analog input channels configured as single-ended inputs or pseudo-differential pairs. The MCP3204 can be configured to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3208 can be configured to provide four pseudo-differential input pairs or eight single-ended inputs. Configuration is done as part of the serial command before each conversion begins. When used in the pseudo-differential mode, each channel pair (i.e., CH0 and CH1, CH2 and CH3 etc.) are programmed as the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ input can range from IN- to (V_{REF} + IN-). The IN- input is limited to ±100mV from the V_{SS} rail. The IN- input can be used to cancel small signal common-mode noise which is present on both the IN+ and IN- inputs.

When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be 000h. If the voltage at IN+ is equal to or greater than {V_{REF} + (IN-) - 1 LSB}, then the output code will be FFFh. If the voltage level at IN- is more than 1 LSB below V_{SS}, then the voltage level at the IN+ input will have to go below V_{SS} to see the 000h output code. Conversely, if IN- is more than 1 LSB above V_{SS}, then the FFFh code will not be seen unless the IN+ input level goes above V_{REF} level.

For the A/D Converter to meet specification, the charge holding capacitor, (C_{SAMPLE}) must be given enough time to acquire a 12-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

In this diagram it is shown that the source impedance (R_S) adds to the internal sampling switch (R_{SS}) impedance, directly affecting the time that is required to charge the capacitor, C_{SAMPLE}. Consequently, larger source impedances increase the offset, gain, and integral linearity errors of the conversion. See Figure 4-2.

4.2 Reference Input

For each device in the family, the reference input (V_{REF}) determines the analog input voltage range. As the reference input is reduced, the LSB size is reduced accordingly. The theoretical digital output code produced by the A/D Converter is a function of the analog input signal and the reference input as shown below.

$$\text{Digital Output Code} = \frac{4096 * V_{IN}}{V_{REF}}$$

where:

V_{IN} = analog input voltage

V_{REF} = reference voltage

When using an external voltage reference device, the system designer should always refer to the manufacturer's recommendations for circuit layout. Any instability in the operation of the reference device will have a direct effect on the operation of the A/D Converter.

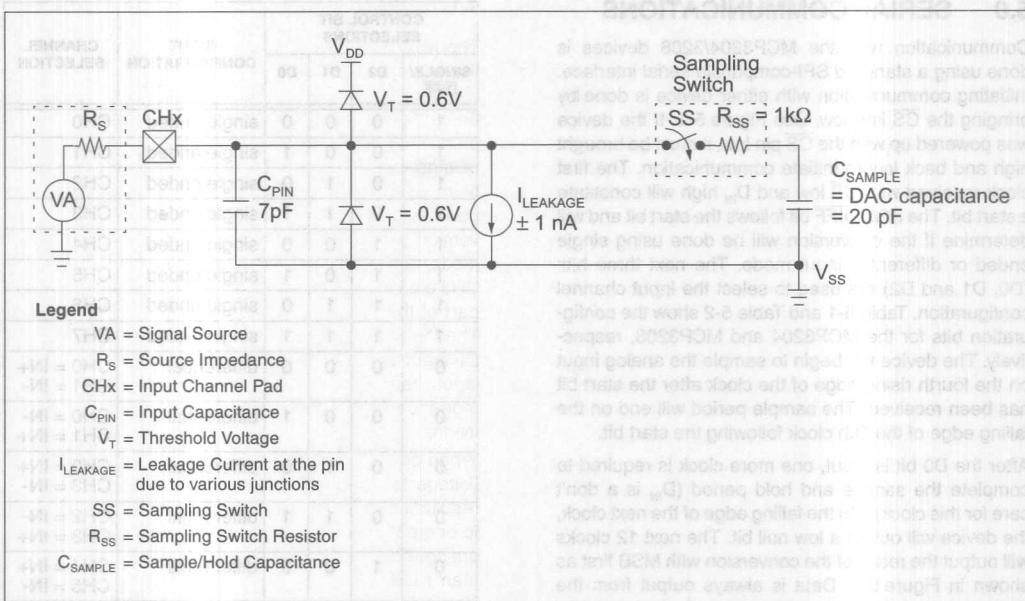


FIGURE 4-1: Analog Input Model

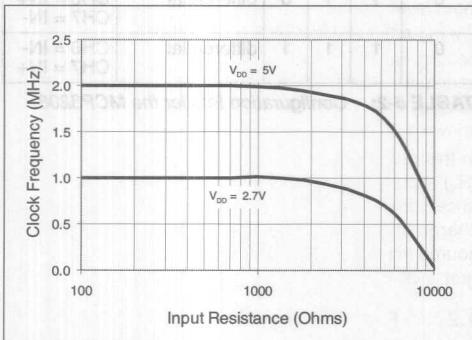


FIGURE 4-2: Maximum Clock Frequency vs. Input resistance (R_S) to maintain less than a 0.1LSB deviation in INL from nominal conditions.

INPUT CHANNEL		REFERENCE		SERIAL PORT	
INPUT	OUTPUT	REF IN	REF OUT	DATA	CLK
0	0	0	0	X	0
1	1	1	1	X	1
2	0	0	0	X	1
3	1	1	1	X	1
4	0	0	0	X	0
5	1	1	1	X	0
6	0	0	0	X	0
7	1	1	1	X	0

MCP3204/3208

5.0 SERIAL COMMUNICATIONS

Communication with the MCP3204/3208 devices is done using a standard SPI-compatible serial interface. Initiating communication with either device is done by bringing the CS line low. See Figure 5-1. If the device was powered up with the CS pin low, it must be brought high and back low to initiate communication. The first clock received with CS low and D_{IN} high will constitute a start bit. The SGL/DIFF bit follows the start bit and will determine if the conversion will be done using single ended or differential input mode. The next three bits (D0, D1 and D2) are used to select the input channel configuration. Table 5-1 and Table 5-2 show the configuration bits for the MCP3204 and MCP3208, respectively. The device will begin to sample the analog input on the fourth rising edge of the clock after the start bit has been received. The sample period will end on the falling edge of the fifth clock following the start bit.

After the D0 bit is input, one more clock is required to complete the sample and hold period (D_{IN} is a don't care for this clock). On the falling edge of the next clock, the device will output a low null bit. The next 12 clocks will output the result of the conversion with MSB first as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 12 data bits have been transmitted and the device continues to receive clocks while the CS is held low, the device will output the conversion result LSB first as shown in Figure 5-2. If more clocks are provided to the device while CS is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring CS low and clock in leading zeros on the D_{IN} line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.1 for more details on using the MCP3204/3208 devices with hardware SPI ports.

CONTROL BIT SELECTIONS				INPUT CONFIGURATION	CHANNEL SELECTION
SINGLE/ DIFF	D2*	D1	D0		
1	X	0	0	single ended	CH0
1	X	0	1	single ended	CH1
1	X	1	0	single ended	CH2
1	X	1	1	single ended	CH3
0	X	0	0	differential CH0 = IN+ CH1 = IN-	CH0 = IN+ CH1 = IN-
0	X	0	1	differential CH0 = IN- CH1 = IN+	CH0 = IN- CH1 = IN+
0	X	1	0	differential CH2 = IN+ CH3 = IN-	CH2 = IN+ CH3 = IN-
0	X	1	1	differential CH2 = IN- CH3 = IN+	CH2 = IN- CH3 = IN+
0	X	0	0	differential CH4 = IN+ CH5 = IN-	CH4 = IN+ CH5 = IN-
0	X	1	0	differential CH6 = IN+ CH7 = IN-	CH6 = IN+ CH7 = IN-
0	X	1	1	differential CH6 = IN- CH7 = IN+	CH6 = IN- CH7 = IN+

*D2 is don't care for MCP3204

TABLE 5-1: Configuration Bits for the MCP3204.

CONTROL BIT SELECTIONS				INPUT CONFIGURATION	CHANNEL SELECTION
SINGLE/ DIFF	D2	D1	D0		
1	0	0	0	single ended	CH0
1	0	0	1	single ended	CH1
1	0	1	0	single ended	CH2
1	0	1	1	single ended	CH3
1	1	0	0	single ended	CH4
1	1	0	1	single ended	CH5
1	1	1	0	single ended	CH6
1	1	1	1	single ended	CH7
0	0	0	0	differential CH0 = IN+ CH1 = IN-	CH0 = IN+ CH1 = IN-
0	0	0	1	differential CH0 = IN- CH1 = IN+	CH0 = IN- CH1 = IN+
0	0	1	0	differential CH2 = IN+ CH3 = IN-	CH2 = IN+ CH3 = IN-
0	0	1	1	differential CH2 = IN- CH3 = IN+	CH2 = IN- CH3 = IN+
0	1	0	0	differential CH4 = IN+ CH5 = IN-	CH4 = IN+ CH5 = IN-
0	1	0	1	differential CH6 = IN+ CH7 = IN-	CH6 = IN+ CH7 = IN-
0	1	1	0	differential CH6 = IN- CH7 = IN+	CH6 = IN- CH7 = IN+
0	1	1	1	differential CH6 = IN- CH7 = IN+	CH6 = IN- CH7 = IN+

TABLE 5-2: Configuration Bits for the MCP3208.



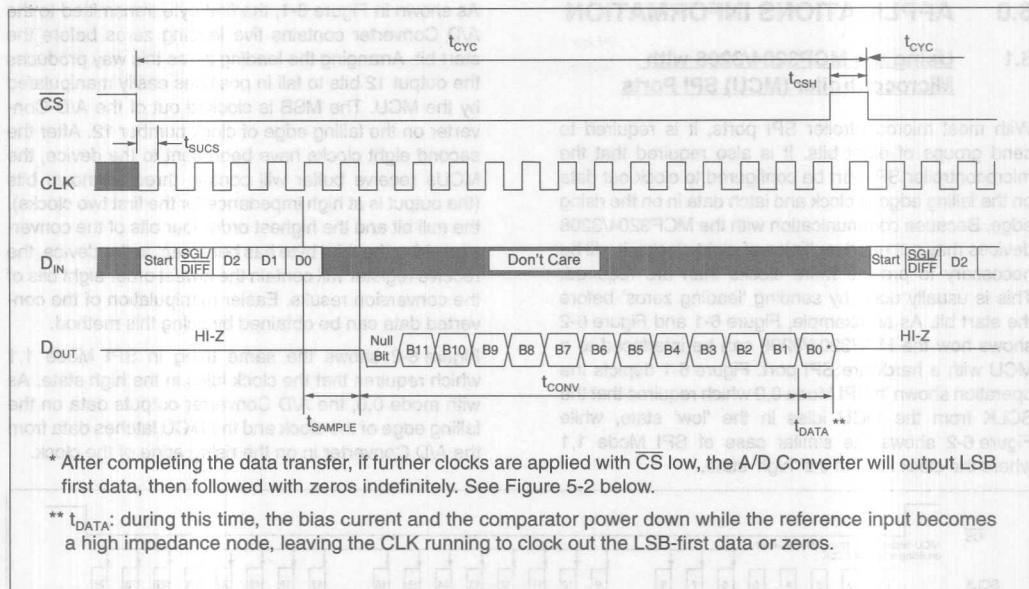


FIGURE 5-1: Communication with the MCP3204 or MCP3208.

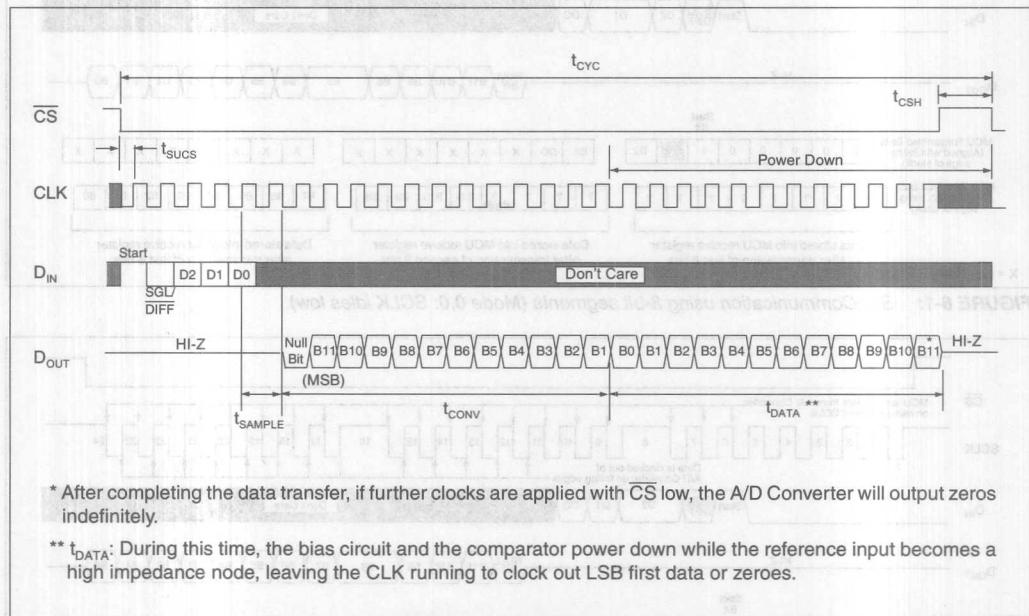
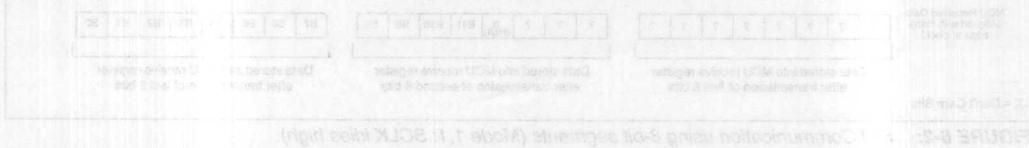


FIGURE 5-2: Communication with MCP3204 or MCP3208 in LSB First Format.



6.1 Using the MCP3204/3208 with Microcontroller (MCU) SPI Ports

With most microcontroller SPI ports, it is required to send groups of eight bits. It is also required that the microcontroller SPI port be configured to clock out data on the falling edge of clock and latch data in on the rising edge. Because communication with the MCP3204/3208 devices may not need multiples of eight clocks, it will be necessary to provide more clocks than are required. This is usually done by sending 'leading zeros' before the start bit. As an example, Figure 6-1 and Figure 6-2 shows how the MCP3204/3208 can be interfaced to a MCU with a hardware SPI port. Figure 6-1 depicts the operation shown in SPI Mode 0,0 which requires that the SCLK from the MCU idles in the 'low' state, while Figure 6-2 shows the similar case of SPI Mode 1,1 where the clock idles in the 'high' state.

A/D Converter contains five leading zeros before the start bit. Arranging the leading zeros this way produces the output 12 bits to fall in positions easily manipulated by the MCU. The MSB is clocked out of the A/D Converter on the falling edge of clock number 12. After the second eight clocks have been sent to the device, the MCUs receive buffer will contain three unknown bits (the output is at high impedance for the first two clocks), the null bit and the highest order four bits of the conversion. After the third byte has been sent to the device, the receive register will contain the lowest order eight bits of the conversion results. Easier manipulation of the converted data can be obtained by using this method.

Figure 6-2 shows the same thing in SPI Mode 1,1 which requires that the clock idles in the high state. As with mode 0,0, the A/D Converter outputs data on the falling edge of the clock and the MCU latches data from the A/D Converter in on the rising edge of the clock.

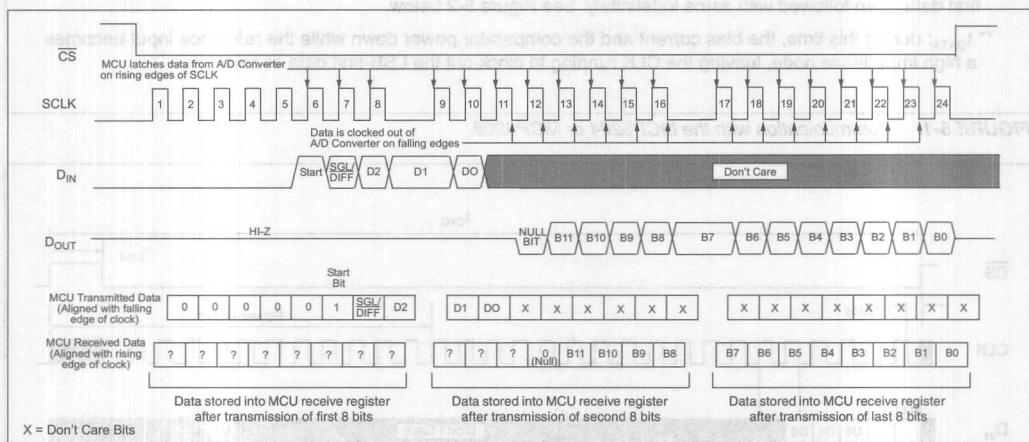


FIGURE 6-1: SPI Communication using 8-bit segments (Mode 0,0: SCLK idles low).

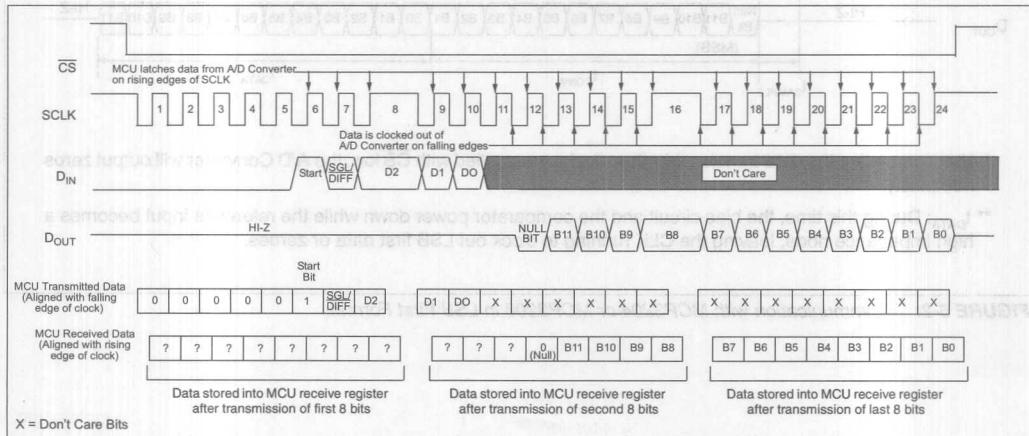


FIGURE 6-2: SPI Communication using 8-bit segments (Mode 1,1: SCLK idles high).

6.2 Maintaining Minimum Clock Speed

When the MCP3204/3208 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample capacitor while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample capacitor for at least 1.2ms after the sample period has ended. This means that the time between the end of the sample period and the time that all 12 data bits have been clocked out must not exceed 1.2ms (effective clock frequency of 10kHz). Failure to meet this criterion may induce linearity errors into the conversion outside the rated specifications. It should be noted that during the entire conversion cycle, the A/D Converter does not require a constant clock speed or duty cycle, as long as all timing specifications are met.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the A/D Converter is not a low impedance source, it will have to be buffered or inaccurate conversion results may occur. See Figure 4-2. It is also recommended that a filter be used to eliminate any signals that may be aliased back in to the conversion results. This is illustrated in Figure 6-3 where an op amp is used to drive the analog input of the MCP3204/3208. This amplifier provides a low impedance source for the converter input and a low pass filter, which eliminates unwanted high frequency noise.

Low pass (anti-aliasing) filters can be designed using Microchip's free interactive **FilterLab™** software. **FilterLab** will calculate capacitor and resistors values, as well as determine the number of poles that are required for the application. For more information on filtering signals, see the application note AN699 "Anti-Aliasing Analog Filters for Data Acquisition Systems."

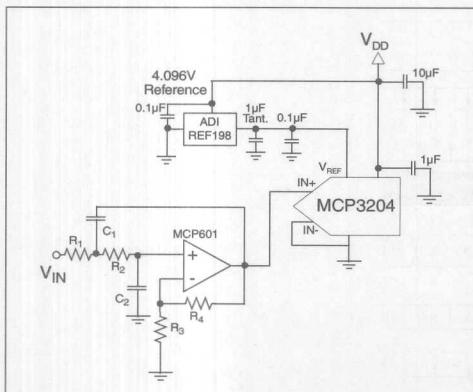


FIGURE 6-3: The MCP601 Operational Amplifier is used to implement a 2nd order anti-aliasing filter for the signal being converted by the MCP3204.

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of $1\mu\text{F}$ is recommended.

Digital and analog traces should be separated as much as possible on the board and no traces should run underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating return current paths and associated errors. See Figure 6-4. For more information on layout tips when using A/D Converters, refer to AN688 "Layout Tips for 12-Bit A/D Converter Applications".

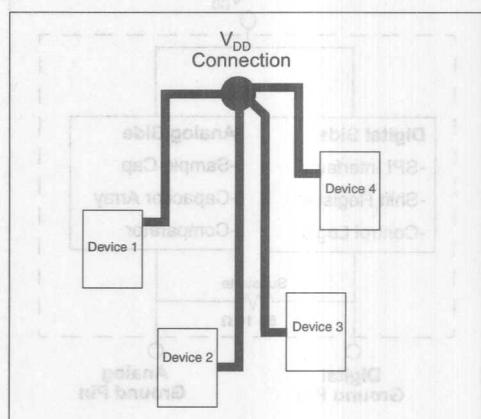


FIGURE 6-4: V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.

MCP3204/3208

6.5 Utilizing the Digital and Analog Ground Pins

The MCP3204/3208 devices provide both digital and analog ground connections to provide another means of noise reduction. As shown in Figure 6-5, the analog and digital circuitry is separated internal to the device. This reduces noise from the digital portion of the device being coupled into the analog portion of the device. The two grounds are connected internally through the substrate which has a resistance of 5 - 10 Ω .

If no ground plane is utilized, then both grounds must be connected to V_{SS} on the board. If a ground plane is available, both digital and analog ground pins should be connected to the analog ground plane. If both an analog and a digital ground plane are available, both the digital and the analog ground pins should be connected to the analog ground plane. Following these steps will reduce the amount of digital noise from the rest of the board being coupled into the A/D Converter.

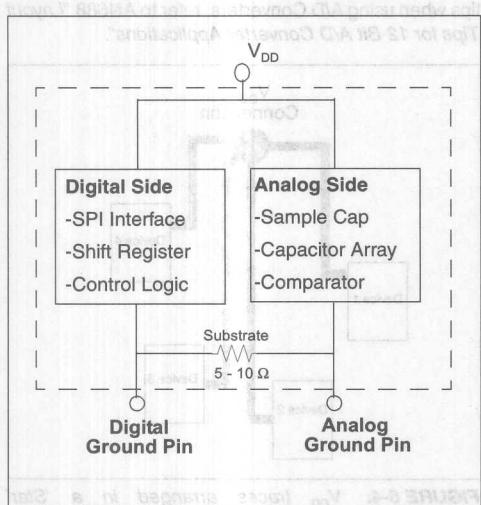


FIGURE 6-5: Separation of Analog and Digital Ground Pins.

MCP3204 PRODUCT IDENTIFICATION SYSTEMS

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

MCP3204 - G T /P	Package:	P = PDIP (14 lead) SL = SOIC (150 mil Body), 14 lead ST = TSSOP, 14 lead (C Grade only)
	Temperature Range:	I = -40°C to +85°C
	Performance Grade:	B = ±1 LSB INL (TSSOP not available in this grade) C = ±2 LSB INL
	Device:	MCP3204 = 4-Channel 12-Bit Serial A/D Converter MCP3204T = 4-Channel 12-Bit Serial A/D Converter on tape and reel (SOIC and TSSOP packages only)

MCP3208 PRODUCT IDENTIFICATION SYSTEMS

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

MCP3208 - G T /P	Package:	P = PDIP (16 lead) SL = SOIC (150 mil Body), 16 lead
	Temperature Range:	I = -40°C to +85°C
	Performance Grade:	B = ±1 LSB INL (TSSOP not available in this grade) C = ±2 LSB INL
	Device:	MCP3208 = 8-Channel 12-Bit Serial A/D Converter MCP3208T = 8-Channel 12-Bit Serial A/D Converter on tape and reel (SOIC packages only)

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

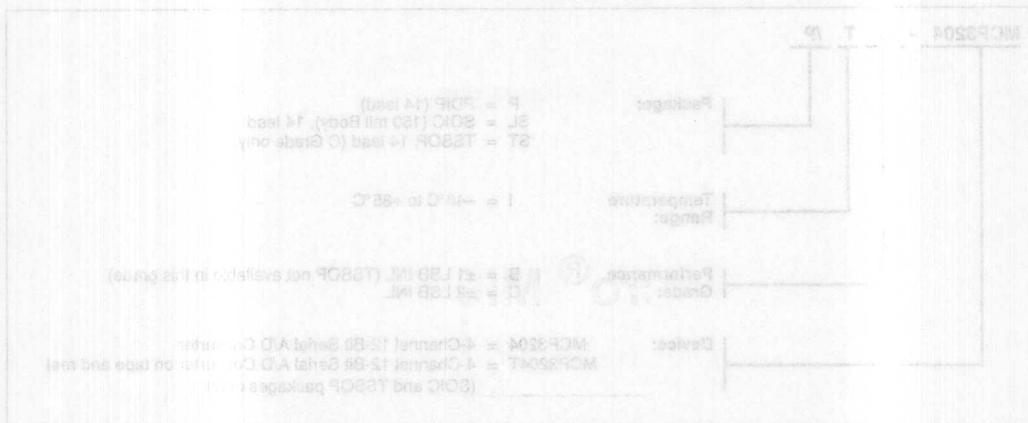
1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277. After September 1, 1999, (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System

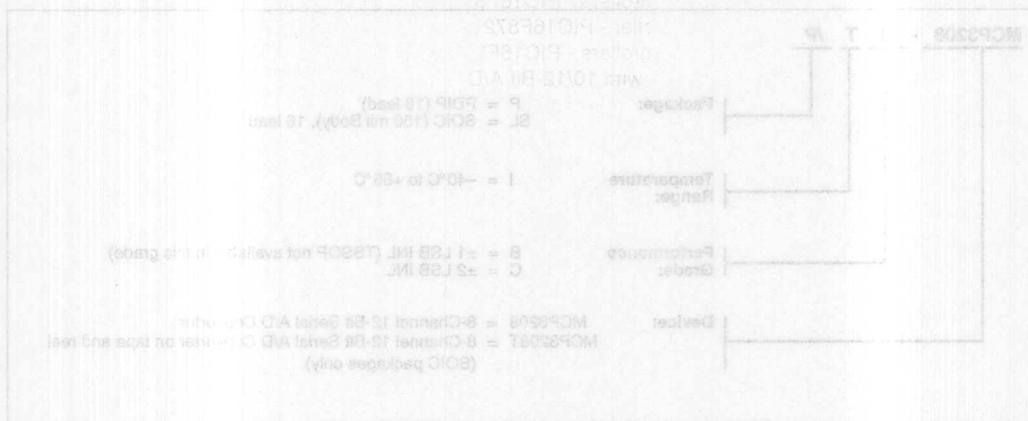
Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

To obtain detailed information on a specific feature or function refer to the detailed descriptions in the following sections.



MCP2508 + QIN2508 IDENTIFICATION SYSTEMS

To obtain detailed information on a specific feature or function refer to the detailed descriptions in the following sections.



Serial Bus Pin-outs

Please note that the serial bus interface is a bidirectional bus. It is shared by several devices. The bus master may issue a slave device addressing unit operation. The bus slave will respond to the bus master. If the bus master initiates a bus transfer, the bus slave must wait until the bus transfer is completed. When the bus transfer is completed, the bus slave may initiate a bus transfer.

1. You must have a valid serial option.
2. The PIC16F877A requires a minimum of 1.024 MHz clock.
3. The MCP2508 is a 16-bit SPI safety ADC converter.

Please refer to the PIC16F877A data sheet for detailed information on how to implement a serial bus interface.

For further information, refer to the Application Note.

Refer to the Microchip Application Note at www.microchip.com.

SECTION 4

PICmicro® MICROCONTROLLER PRODUCT BRIEFS

High-Performance Microcontrollers with 10-Bit A/D - PIC18Cxx2.....	4-1
28/40-Pin, 8-Bit CMOS Microcontrollers w/ 12-Bit A/D - PIC16C77X	4-3
28/40-pin 8-Bit CMOS FLASH Microcontrollers - PIC16F870/871	4-7
28-Pin, 8-Bit CMOS FLASH Microcontroller - PIC16F872	4-11
28/40-pin 8-Bit CMOS FLASH Microcontrollers - PIC16F87X	4-13
18/20-Pin, 8-Bit CMOS Microcontrollers with 10/12-Bit A/D - PIC16C717/770/771	4-17

4 SECTIÖN MICROCHIP JOINTER PRODUCTS BRIEFS

4-1	18V50-Pin, 8-Bit CMOS Microcontroller with 10-Bit A/D - PIC18Cxx5
4-2	28V40-Pin, 8-Bit CMOS FLASH Microcontroller - PIC18F8125
4-3	28V40-Pin, 8-Bit CMOS FLASH Microcontroller - PIC18F825
4-4	28V40-Pin, 8-Bit CMOS FLASH Microcontroller - PIC18F82X
4-5	28V40-Pin, 8-Bit CMOS FLASH Microcontroller - PIC18F84X

Microchip Technology Inc.



MICROCHIP

PIC18CXX2

PIC18CXX2

High-Performance Microcontrollers with 10-Bit A/D

High Performance RISC CPU:

- C-compiler optimized architecture/instruction set
 - Code compatible with Midrange instruction set
- Linear program memory addressing to 2M bytes
- Linear data memory addressing to 4K bytes

Device	On-Chip Program Memory	On-Chip RAM (bytes)
	EPROM (bytes)	# Single Word Instructions
PIC18C242	16K	8192
PIC18C252	32K	16384
PIC18C442	16K	8192
PIC18C452	32K	16384

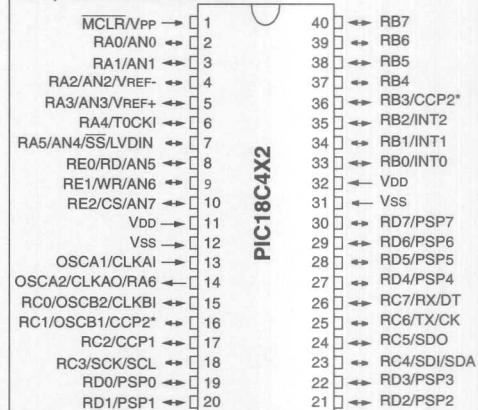
- Up to 10 MIPS operation:
 - DC - 40 MHz osc./clock input
 - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- **Timer0** module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- **Timer1** module: 16-bit timer/counter (time-based for capture/compare). Secondary oscillator as clock input option
- **Timer2** module: 8-bit timer/counter with 8-bit period register (time-based for PWM)
- **Timer3** module: 16-bit timer/counter (alternate time-base for capture/compare). Secondary oscillator as clock input option
- Two **Capture/Compare/PWM** (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (Tcy/16)
 - Compare is 16-bit, max. resolution 100 ns (Tcy)
 - PWM output: PWM resolution is 1- to 10-bit. Max. PWM freq. @: 8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- **Master Synchronous Serial Port** (MSSP) module. Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I²C™ master and slave mode
- **Addressable USART** (AUSART) module:
 - Supports interrupt on Address bit
- **Parallel Slave Port** (PSP) module

Pin Diagrams

DIP, Windowed CERDIP



* RB3 is the alternate pin for the CCP2 pin multiplexing.

NOTE: Pin compatible with 40-pin PIC16C7X devices

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during sleep
 - DNL = ±1 Lsb, INL = ±1 Lsb
- Programmable Low-Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
- Programmable Brown-out Reset (BOR)

Special Microcontroller Features:

- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options including:
 - 4X Phase Lock Loop (of primary oscillator)
 - Secondary Oscillator (32 kHz) clock input
- In-Circuit Serial Programming (ICSP™) via two pins

CMOS Technology:

- Low-power, high-speed EPROM technology
- Fully static design
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption



MICROCHIP

PIC16C77X

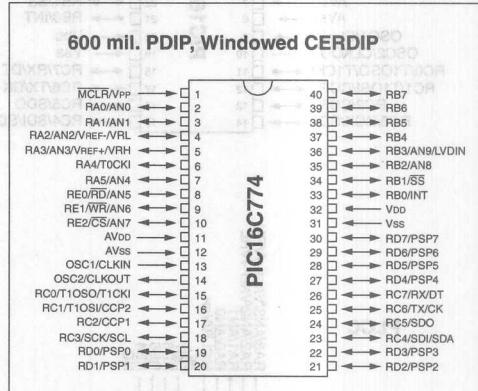
PIC16C77X

28/40-Pin, 8-Bit CMOS Microcontrollers w/ 12-Bit A/D

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 4K x 14 words of Program Memory,
256 x 8 bytes of Data Memory (RAM)
- Interrupt capability (up to 14 internal/external interrupt sources)
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS EPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP™)
- Wide operating voltage range: 2.5V to 5.5V
- High Sink/Source Current 25/25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
 - < 2 mA @ 5V, 4 MHz
 - 22.5 μ A typical @ 3V, 32 kHz
 - < 1 μ A typical standby current

Pin Diagram



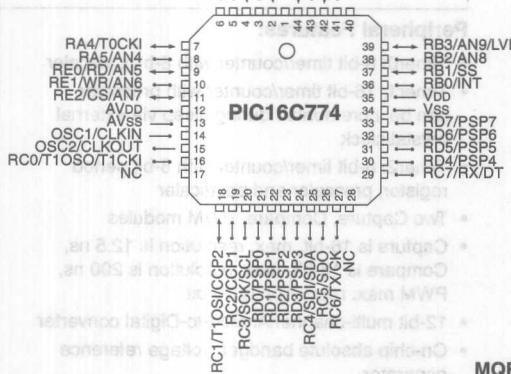
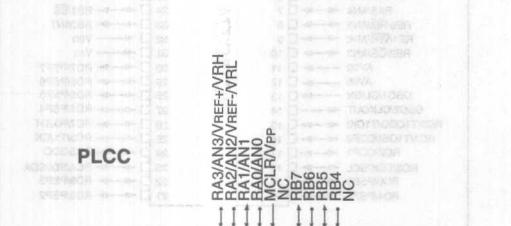
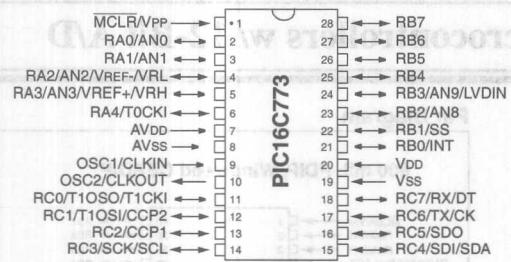
Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
- Capture is 16-bit, max. resolution is 12.5 ns, Compare is 16-bit, max. resolution is 200 ns, PWM max. resolution is 10-bit
- 12-bit multi-channel Analog-to-Digital converter
- On-chip absolute bandgap voltage reference generator
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I²C™
- Universal Synchronous Asynchronous Receiver Transmitter, supports high/low speeds and 9-bit address mode (USART/SCI)
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls
- Programmable Brown-out detection circuitry for Brown-out Reset (BOR)
- Programmable Low-voltage detection circuitry

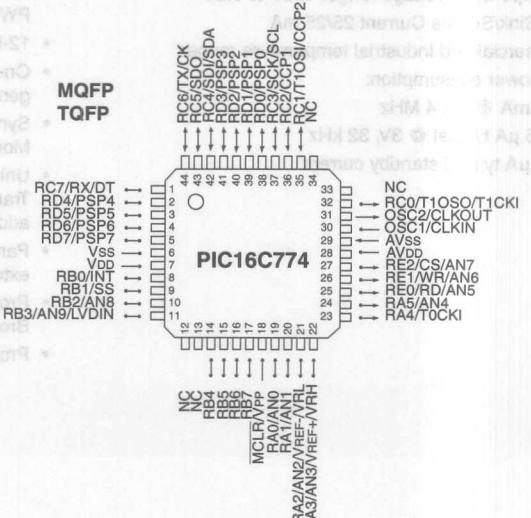
PIC16C77X

Pin Diagrams

300 mil. SDIP, SOIC, Windowed CERDIP, SSOP



**MQFP
TQFP**



Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16C773	PIC16C774
Operating Frequency	DC - 20 MHz	DC - 20 MHz
Resets (and Delays)	POR, BOR, MCLR, WDT (PWRT, OST)	POR, BOR, MCLR, WDT (PWRT, OST)
Program Memory (14-bit words)	4K	4K
Data Memory (bytes)	256	256
Interrupts	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3
Capture/Compare/PWM modules	2	2
Serial Communications	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP
12-bit Analog-to-Digital Module	6 input channels	10 input channels
Instruction Set	35 Instructions	35 Instructions

PIC16C77X

NOTES:

Notes	Ref ID	Description	Notes
DC-50 MHz	DC-50 MHz	DC-50 MHz	DC-50 MHz
Power MOSFET	Power MOSFET	Power MOSFET	Power MOSFET
4K	4K	4K	4K
256	256	256	256
14	14	14	14
Power A,B,C	Power A,B,C	Power A,B,C	Power A,B,C
3	3	3	3
5	5	5	5
MSB TRA8T	MSB TRA8T	MSB TRA8T	MSB TRA8T
—	—	—	—
10 bits	10 bits	10 bits	10 bits
28 pins	28 pins	28 pins	28 pins
DS30098A	DS30098A	DS30098A	DS30098A



MICROCHIP

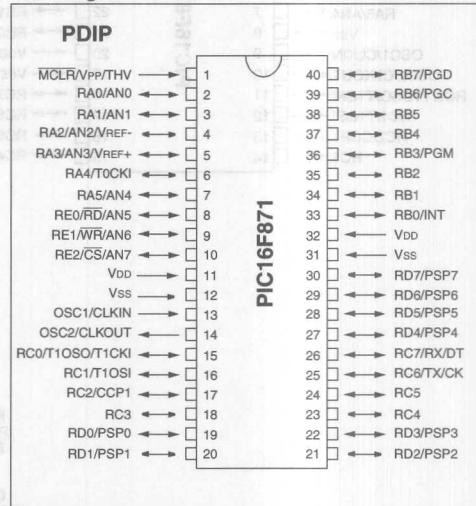
PIC16F870/871

28/40-pin 8-Bit CMOS FLASH Microcontrollers

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 2K x 14 words of FLASH Program Memory
128 x 8 bytes of Data Memory (RAM)
64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16CXXX 28 and 40-pin devices
- Interrupt capability (up to 11 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 20 μ A typical @ 3V, 32 kHz
 - < 1 μ A typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- One Capture, Compare, PWM module
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

PIC16F870/871

Pin Diagrams

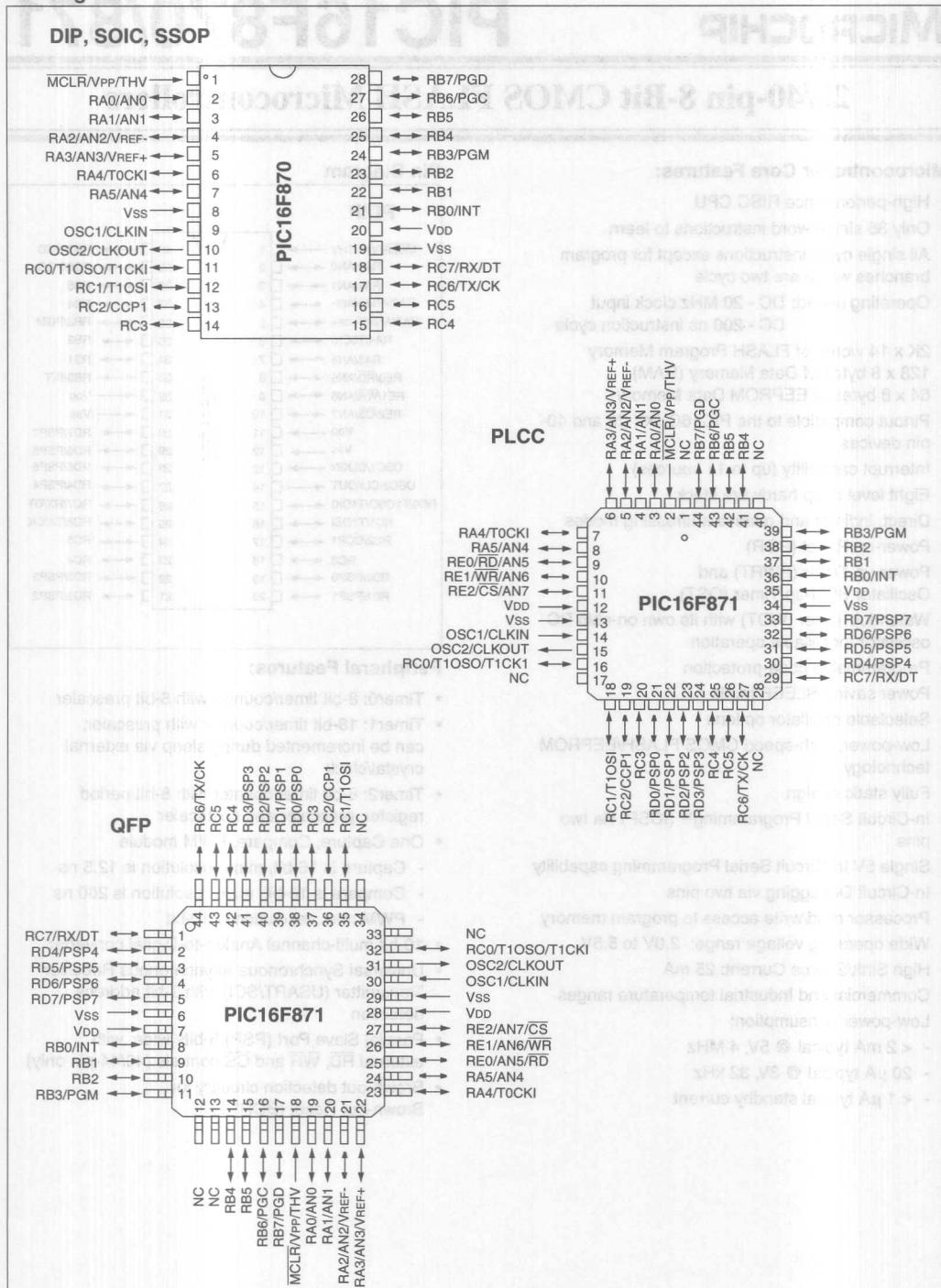
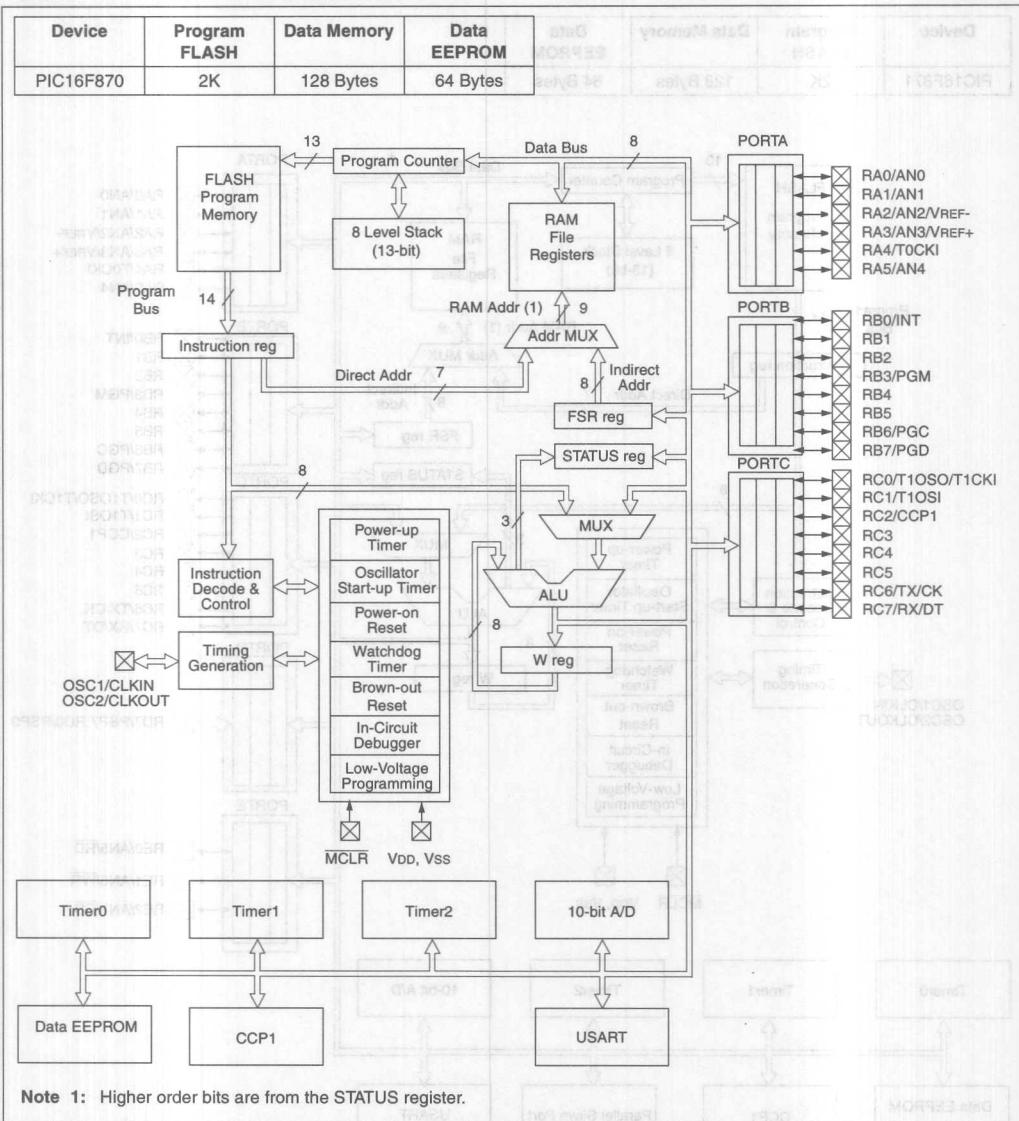
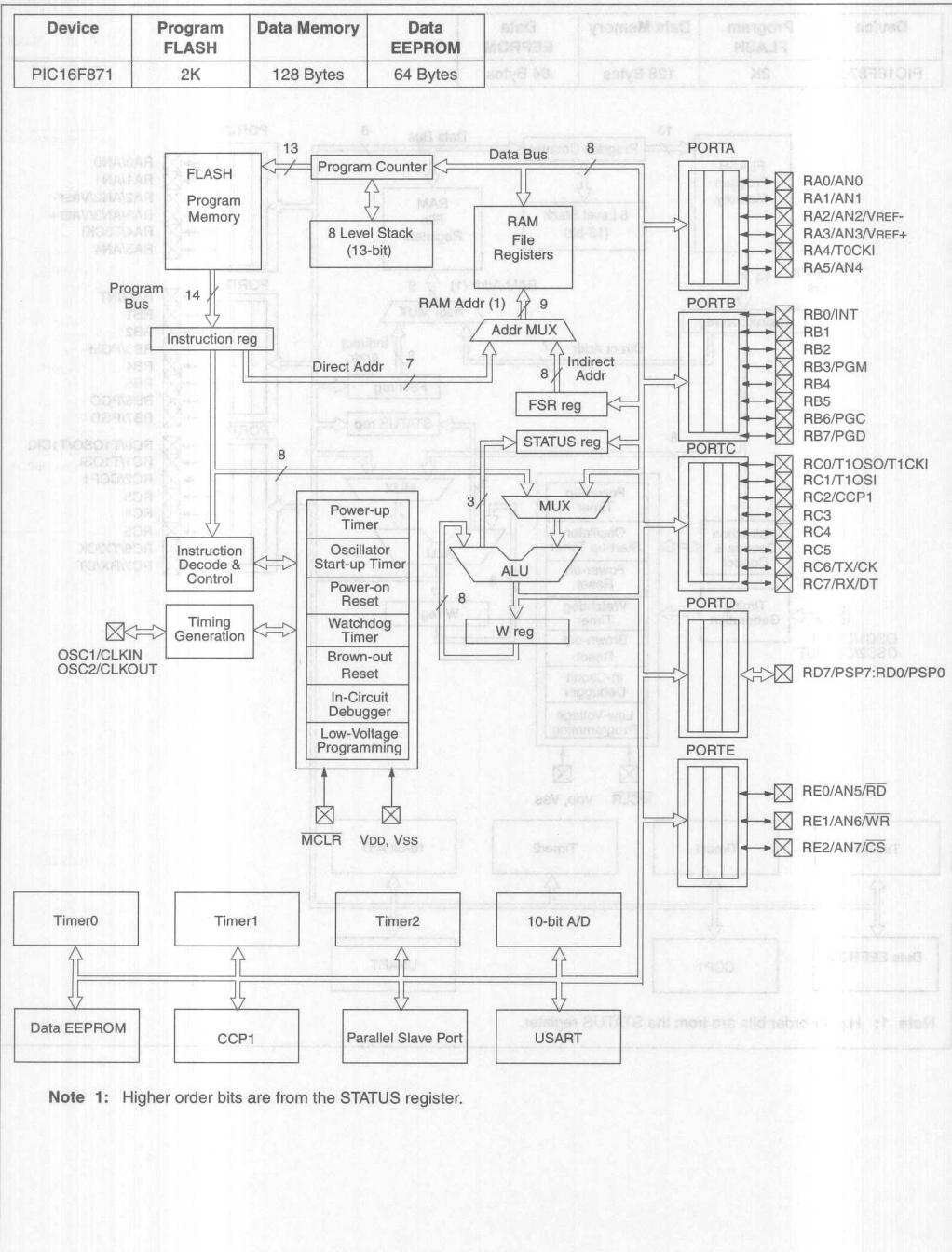


FIGURE 1: PIC16F870 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.

FIGURE 2: PIC16F871 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.



MICROCHIP

PIC16F872

28-Pin, 8-Bit CMOS FLASH Microcontroller

Devices Included in this Data Sheet:

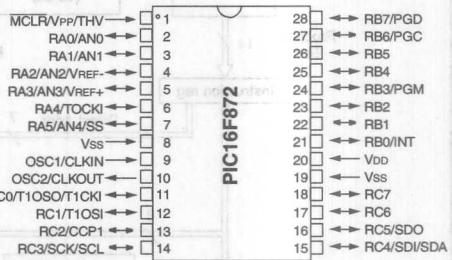
- PIC16F872

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 2K x 14 words of FLASH Program Memory
128 x 8 bytes of Data Memory (RAM)
64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C72A
- Interrupt capability (up to 10 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram

DIP, SOIC, SSOP

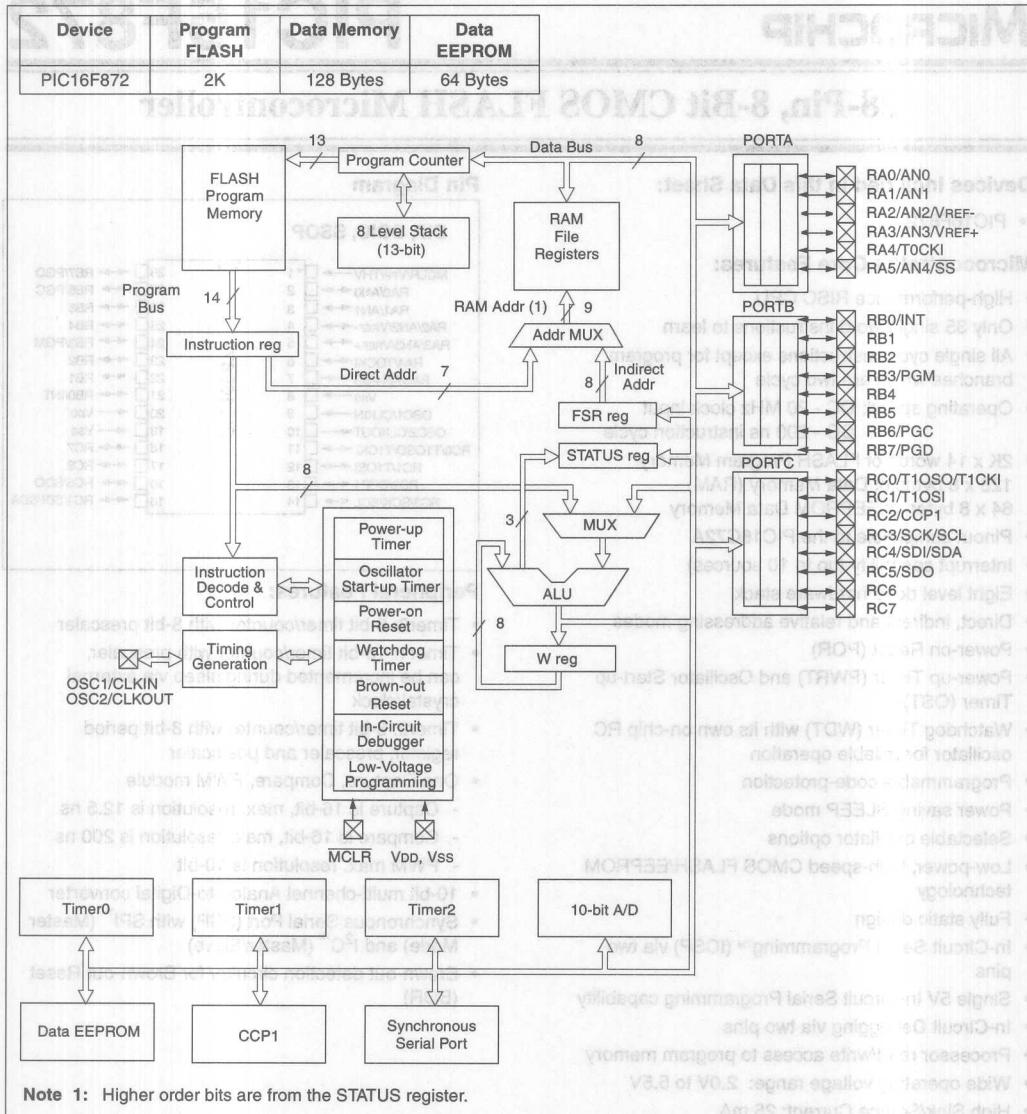


Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- One Capture, Compare, PWM module
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I²C™ (Master/Slave)
- Brown-out detection circuitry for Brown-out Reset (BOR)

PIC16F872

FIGURE 1: PIC16F872 BLOCK DIAGRAM





MICROCHIP

PIC16F87X

28/40-pin 8-Bit CMOS FLASH Microcontrollers

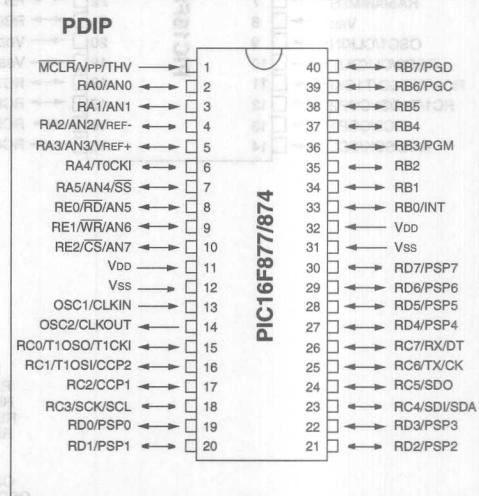
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM data memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



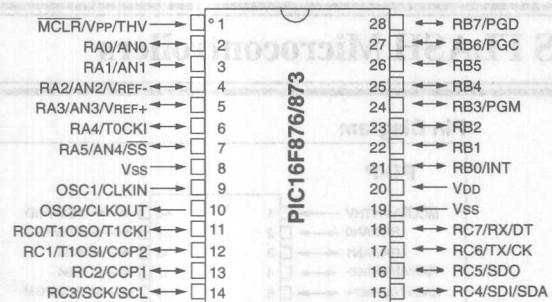
Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

PIC16F87X

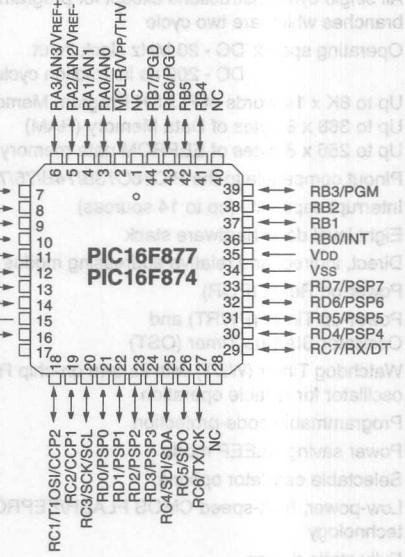
Pin Diagrams

DIP, SOIC

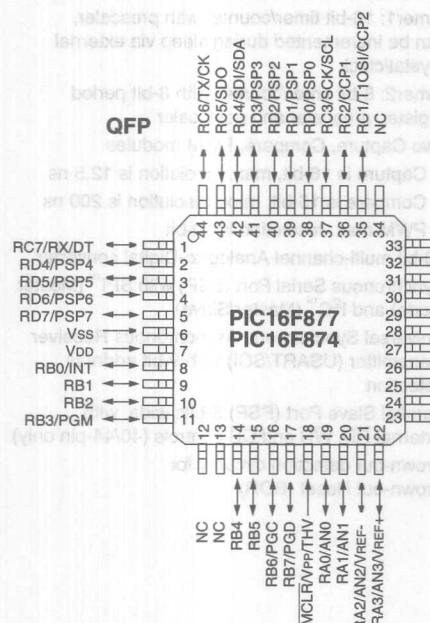


PIC16F876/873

PLCC



PIC16F877
PIC16F874



PIC16F877
PIC16F874

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz			
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions

PIC16F87X

NOTES:



MICROCHIP

PIC16C717/770/771

18/20-Pin, 8-Bit CMOS Microcontrollers with 10/12-Bit A/D

Microcontroller Core Features:

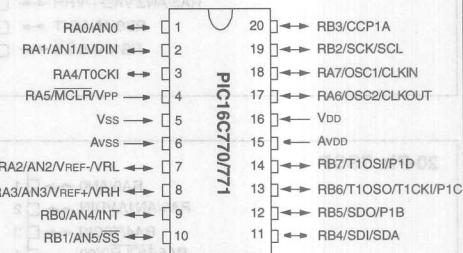
- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle

Device	Memory		Pins	A/D Resolution	A/D Channels
	Program x14	Data x8			
PIC16C717	2K	256	18, 20	10 bits	6
PIC16C770	2K	256	20	12 bits	6
PIC16C771	4K	256	20	12 bits	6

- Interrupt capability (up to 10 internal/external interrupt sources)
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Selectable oscillator options:
 - INTRC - Internal RC, dual speed (4MHz and 37KHz), with or without clock output
 - ER - External resistor, dual speed (user selectable frequency and 37KHz), with or without clock output
 - EC - External clock
 - HS - High speed crystal/resonator
 - XT - Crystal/resonator
 - LP - Low power crystal
- Low-power, high-speed CMOS EPROM technology
- In-Circuit Serial Programming™ (ISCP)
- Wide operating voltage range: 2.5V to 5.5V
- 15 I/O pins with control for:
 - Direction (15)
 - Port/analog input (6)
 - PORTB interrupt on change (8)
 - PORTB weak pull-up (8)
 - High voltage open drain (1)
- Commercial and Industrial temperature ranges
- Low-power consumption:
 - < 2 mA @ 5V, 4 MHz
 - 22.5 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram

20-Pin PDIP, SOIC, SSOP



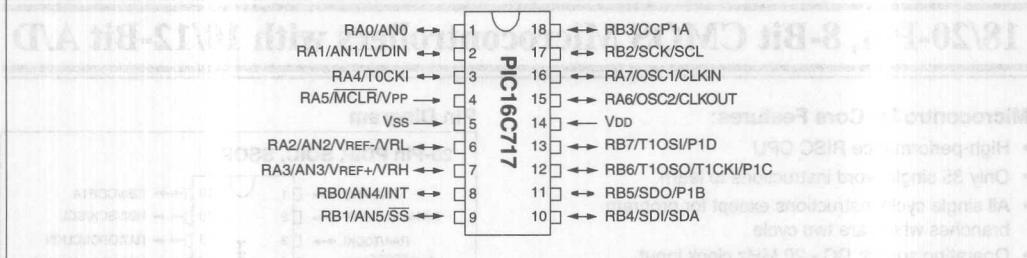
Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM (ECCP) module
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
 - Enhanced PWM
- Analog-to-Digital converter:
 - PIC16C717 10-bit resolution
 - PIC16C770/771 12-bit resolution
- On-chip absolute bandgap voltage reference generator
- Programmable Brown-out Detection (PBOD) circuitry for Brown-out Reset (BOR)
- Programmable Low-Voltage Detection (PLVD) circuitry
- Master Synchronous Serial Port (MSSP) with two modes of operation:
 - 3-wire SPI (supports all 4 SPI modes)
 - I²C compatible including master mode support

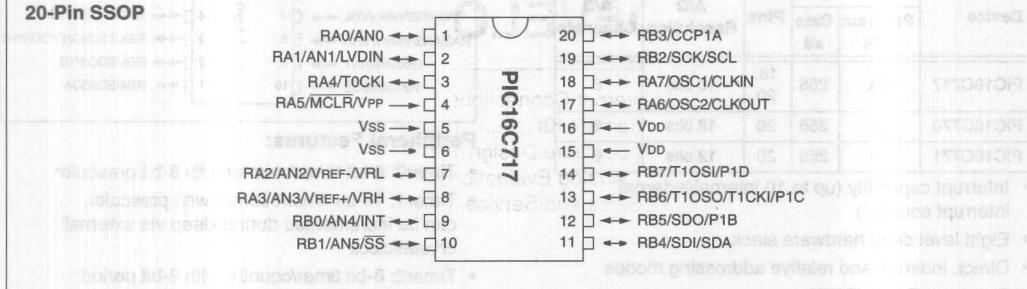
PIC16C717/770/771

PIN DIAGRAMS

18-Pin PDIP, SOIC



20-Pin SSOP



Key Features

**PICmicro™ Mid-Range Reference Manual
(DS33023)**

	PIC16C717	PIC16C770	PIC16C771
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
Resets (and Delays)	POR, BOR, MCLR, WDT (PWRT, OST)	POR, BOR, MCLR, WDT (PWRT, OST)	POR, BOR, MCLR, WDT (PWRT, OST)
Program Memory (14-bit words)	2K	2K	4K
Data Memory (bytes)	256	256	256
Interrupts	10	10	10
I/O Ports	Ports A,B	Ports A,B	Ports A,B
Timers	3	3	3
Enhanced Capture/Compare/PWM (ECCP) modules	1	1	1
Serial Communications	MSSP	MSSP	MSSP
10-bit Analog-to-Digital Module	6 input channels	—	—
12-bit Analog-to-Digital Module	—	6 input channels	6 input channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions

SECTION 5 DEVELOPMENT SYSTEMS

On-Line Support	Microchip Internet Connections.....	5-1
MCP2510	CAN Development Kit	5-3
FilterLab™	Active Filter Software Design Tool	5-5
MXDEV™ 1	MXDEV™ 1 Analog Evaluation System.....	5-7
	Worldwide Sales and Service	5-12



MICROCHIP



MICROCHIP

ON-LINE SUPPORT

Microchip Internet Connections

On-line Support

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the internet and a web browser, such as Netscape® or Microsoft® Internet Explorer®. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

<ftp://www.microchip.com>

The web site and file transfer site provide a variety of services. Users may download files for the latest development tools, data sheets, application notes, user's guides, articles, and sample programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip press releases
- Technical support section with frequently asked questions
- Design tips
- Device errata
- Job postings
- Microchip consultant program member listing
- Links to other useful web sites related to Microchip products
- Conferences for products, development systems, technical information and more
- Listing of seminars and events

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Hot Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada
and

1-480-786-7302 for the rest of the world

On-Line Support

NOTES:

ON-LINE SUPPORT

Microchip Internet Connection

Systems Information and Upgrades Hot Lines

The Systems Information and Upgrades Hot Line provides general news to general public and development personnel to all Microchip hardware products. This line also provides information on product availability, new announcements, and general news to the media. The Hot Line number is 1-800-288-5848 for U.S. and most of Canada.

1-800-288-5848 for U.S. and most of Canada
800-288-5805 for the rest of the world



On-Line Support

Microchip offers an on-line support service. Microchip Web (WWW) site. This web site is part of Microchip's on-line connection system. To use this site, first you must have access to the Internet via a Web browser, such as Netscape® or Microsoft Internet Explorer®. This site also allows for FTP download from our FTP site.

Connection to the Microchip Support Web Site

The Microchip main page is available by going to our homepage or through our support site at www.microchip.com. The following features are available by going to www.microchip.com:

- Latest Microchip release releases
- Technical support section with frequently asked questions
- Device driver section
- Development tools
- Microchip newsletter containing product news and information
- Forum of other Microchip users who share their knowledge and expertise
- Microchip graphics
- Configuration tool for board development
- Software, technical documentation and more
- Training of Microchip products

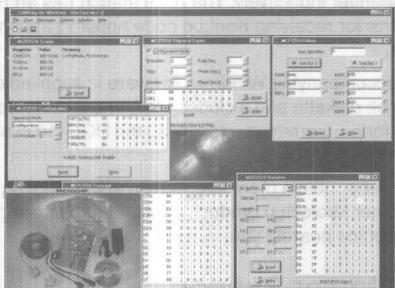


MICROCHIP

MCP2510

MCP2510

CAN Development Kit



The MCP2510 Controller Area Network (CAN) Developer's Kit is ideal for CAN system developers as well as for CAN beginners.

MCP2510 software development is made easy by offering a variety of features to manipulate the functionality of the MCP2510. The MCP2510 CAN Developer's Kit provides the ability to read, display, and modify all registers of the MCP2510 on a

bit-by-bit or a byte-by-byte basis. Included on the target board are PICmicro® sockets, a header to access the required MCP2510 pins, and a prototype area for the user to quickly build and test his own CAN node. Also included are on-board transceivers that have jumper-configureable options to allow different bus setups. In addition, this tool provides the user with an expansion connector for connecting a user-created CAN network. By using this expansion connector in this manner, the PC interface can be used as a simple bus monitor for CAN message traffic.

For CAN beginners, the MCP2510 CAN Developer's Kit can be used as a low-cost method of demonstrating basic input and output functionality by transmitting and receiving CAN messages. Transmitted messages are set up via an easy-to-use Windows® interface. LEDs connected to the MCP2510 transmit and receive pins toggle to show message traffic. Both analog and digital signals can be generated on the target board. These signals are then received by the host PC and displayed in a de-stuffed format for easy identification of message contents. In this manner, basic CAN communication can be demonstrated and understood.

Features:

■ On-board features speed understanding:

- Ability to read, display, and modify all registers
- Ability to manipulate message mask and message filter functions
- Modifications can be done on a bit-by-bit basis or a byte-by-byte basis

■ Aids in development of users' CAN network:

- On-board industry-standard CAN transceivers
- Prototype area for user-defined transceivers that are jumper selectable
- Expansion connector enables users to connect external CAN network and use PC interface as a basic bus monitor
- PICmicro sockets, access to MCP2510 signals and prototype area for quick CAN mode development

■ CAN messages demonstration capability for CAN beginners:

- Easy to create and send CAN messages
- Displays received messages in de-stuffed format
- Target board contains switches and dials to vary message contents
- LEDs toggle on and off to signify CAN message traffic
- Familiar user interface

MCP2510

Ordering Information:

See the Microchip Development Systems Ordering Guide (DS30177) or www.microchip.com for specific part numbers. To order or obtain more information about the MCP2510 CAN Developer's Kit or any other Microchip product, contact the Microchip Sales Office, representative, or distributor nearest you.

Host System Requirements:

PC with 486 or higher processor.

Pentium® recommended

4 MB Memory, 8 MB recommended

2 MB hard disk space,

5 MB recommended

VGA or Super VGA Monitor

Microsoft® Windows 3.1 or greater

Parallel Port

Customer Support:

Microchip maintains a worldwide network of distributors, representatives, local sales offices, Field Application Engineers, and Corporate Application Engineers. Microchip's Internet home page can be reached at: www.microchip.com

System Description:

The KEELOQ Transponder Evaluation Kit Hardware consists of a base station, a transmitter/transponder, a batteryless transponder, and various HCS410 samples. The base station doubles as a programmer and decoder. The base station includes a coil used for generating a magnetic field used to communicate with a transponder inductively. The base station has an RF receiver for receiving KeeLoq code hopping transmissions.

The accompanying Windows software is supplied on a 3.5-inch diskette and includes all the necessary software for programming and testing the HCS410 in all its modes.

To order or obtain more information about KeeLoq or any other Microchip product, contact the Microchip sales office nearest you.

Customer Support:

Microchip maintains a worldwide network of distributors, representatives, local sales offices, Field Application Engineers, and Corporate Application Engineers. Microchip's Internet home page can be reached at: www.microchip.com

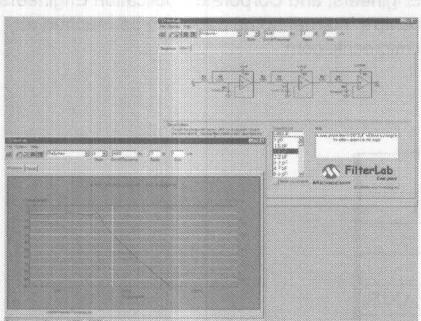
Development Tools from Microchip

MPLAB™	Integrated Development Environment (IDE)
MPLAB-C17	C compiler for PIC17CXXX MCUs
MPLAB-C18	C compiler for PIC18CXXX MCUs
C compiler	Sold by third-party vendors (HI-TECH, IAR, CCS)
MPASM	Universal PICmicro macro-assembler
MPLINK/MPLIB	Linker/Librarian
MPLAB-SIM	Software Simulator
MPLAB-ICD	In-circuit debugger evaluation kit
MPLAB-ICE 2000	Full-featured modular in-circuit emulator
ICEPIC	Low-cost in-circuit emulator
PRO MATE® II	Full-featured, modular device programmer
PICSTART®Plus	Entry-level development kit with programmer



FILTERLAB™

Active Filter Software Design Tool



The difficult job of low-pass, active filter design is made easy with FilterLab software.

FilterLab is an innovative software tool that simplifies active filter design. Available at no cost from Microchip's web site (www.microchip.com), the FilterLab active filter software design tool provides full schematic diagrams of the filter circuit with component values and displays the frequency response.

FilterLab allows the design of low-pass filters up to an 8th order filter with Chebyshev, Bessel or Butterworth responses from frequencies of 0.1 Hz to 10 MHz. Users can select a flat passband or sharp transition from passband to stopband. Options, such as minimum ripple factor, sharp transition and linear phase delay, are available. Once the filter response has been identified, FilterLab generates the frequency response and the circuit. For maximum design flexibility, changes in capacitor values can be implemented to fit the demands of the application. FilterLab will recalculate all values to meet the desired response, allowing real-world values to be substituted or changed as part of the design process.

FilterLab also generates a spice model of the designed filter. Extraction of this model will allow time domain analysis in spice simulations, streamlining the design process.

Further consideration is given to designs used in conjunction with an analog-to-digital converter. A suggested filter can be generated by simply inputting the bit resolution and sample rate via the Anti-Aliasing Wizard. This eliminates erroneous signals folded back into the digital data due to the aliasing effect.

Features:

- Multiple Filter Order and Responses with Gain Option
 - Ability to select Bessel, Butterworth or Chebyshev filter response
 - Up to 8th-order filters can be simulated
 - Circuit diagram and component values given
- Bode Plot with Phase Margin
 - Resultant Bode plot generated
- Circuit Implementation
 - Standard 1 percent resistors
 - Standard capacitor values generate and user adjustable
 - Circuit configuration: Sallen-Key (noninverting) or multiple feedback (inverting)
- Spice Model Generated
 - Spice Model of entire filter generated
 - Allows for streamline of simulations
- Anti-Aliasing Wizard
 - Filter optimization for Analog-to-Digital Converter base on bit resolution and sample rate

Ordering Information:

FilterLab

Devices Supported:

PC

Host System Requirements:

PC with 386 or higher processor.

Pentium® recommended

8 MB Memory, 32 MB recommended

16 MB hard disk space,

20 MB recommended

600 x 800 Monitor

Microsoft® Windows® 95/98

CD-ROM Drive

Customer Support:

Microchip maintains a worldwide network of distributors, representatives, local sales offices, Field Application Engineers, and Corporate Application Engineers.

Microchip's Internet home page can be reached at:

www.microchip.com

Development Tools from Microchip	
MPLAB™	Integrated Development Environment (IDE)
MPLAB-C17	C compiler for PIC17CXXX MCUs
MPLAB-C18	C compiler for PIC18CXXX MCUs
C compiler	Sold by third-party vendors (HI-TECH, IAR, CCS)
MPASM	Universal PICmicro macro-assembler
MPLINK/MPLIB	Linker/Librarian
MPLAB-SIM	Software Simulator
MPLAB-ICD	In-circuit debugger evaluation kit
MPLAB-ICE 2000	Full-featured modular in-circuit emulator
ICEPIC	Low-cost in-circuit emulator
PRO MATE® II	Full-featured, modular device programmer
PICSTART®Plus	Entry-level development kit with programmer



MICROCHIP

MXDEV™
ANALOG
EVALUATION
SYSTEM

MXDEV™ 1

MXDEV™ 1 Analog Evaluation System

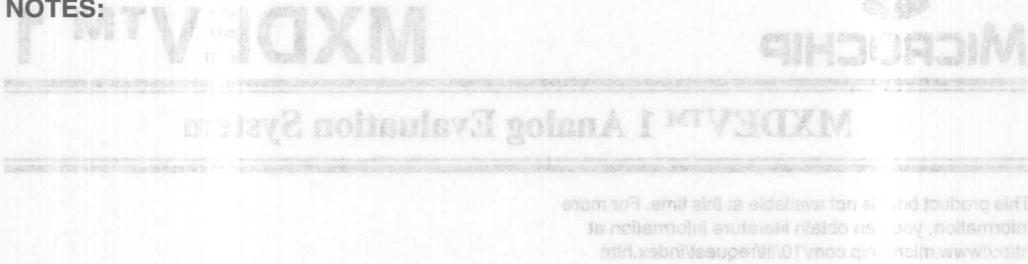
This product brief is not available at this time. For more information, you can obtain literature information at <http://www.microchip.com/10/lit/request/index.htm>

5

Development
Systems

MXDEV™ 1

NOTES:



This document is for reference only. For the latest information, refer to the Microchip website at www.microchip.com.

Development
Board